Modeling Aspect Sentiment Coherency via Local Sentiment Aggregation

Anonymous ACL submission

Abstract

Aspect sentiment coherency is an intriguing yet underexplored topic in the field of aspect-based sentiment classification. This concept reflects the common pattern where adjacent aspects often share similar sentiments. Despite its prevalence, current studies have not fully recognized the potential of modeling aspect sentiment coherency, including its implications in adversarial defense. To model aspect sentiment coherency, we propose a novel local sentiment aggregation (`LSA`) paradigm based on constructing a differential-weighted sentiment aggregation window. We have rigorously evaluated our model through experiments, and the results affirm the proficiency of `LSA` in terms of aspect coherency prediction and aspect sentiment classification. For instance, it outperforms existing models and achieves state-of-the-art sentiment classification performance across five public datasets. Furthermore, we demonstrate the promising ability of `LSA` in ABSC adversarial defense, thanks to its sentiment coherency modeling. To encourage further exploration and application of this concept, we have made our code publicly accessible. This will provide researchers with a valuable tool to delve into sentiment coherency modeling in future research.

1 Introduction

Aspect-based sentiment classification (Pontiki et al., 2014, 2015, 2016) (ABSC) aims to identify sentiments associated with specific aspects within a text, as highlighted in several studies (Ma et al., 2017; Fan et al., 2018; Zhang et al., 2019; Yang et al., 2021). In this work, we make efforts to address an intriguing problem within ABSC that has been overlooked in existing research, i.e., "***aspect sentiment coherency***", which focuses on modeling aspects that share similar sentiments. For instance, in the sentence "*This laptop has a lot of storage, and so does the battery capacity,*" where '*storage*' and '*battery capacity*' aspects both contain positive sentiments. We show more examples of aspect sentiment coherency in Fig. 1 and the case study section.

The study of aspect sentiment coherency has not been investigated in existing research. Yet, some strides have been made on a similar topic, namely sentiment dependency. These approaches, featured in several studies (Zhang et al., 2019; Huang and Carley, 2019; Phan and Ogunbona, 2020), hypothesize that sentiments of aspects may be dependent and usually leverage syntax trees to reveal potential sentiment dependencies between aspects. However, sentiment dependency remains a somewhat ambiguous concept in the current research landscape. Furthermore, previous methods (Zhou et al., 2020; Zhao et al., 2020; Tang et al., 2020; Li et al., 2021a,a) tend to model context topological dependency (e.g., context syntax structure) rather than sentiment dependency directly. These techniques are resource and computation-intensive and can suffer from token-node misalignment caused by conflicts in tokenization methods in syntax tree construction.

As a further contribution to current ABSC research, we propose aspect sentiment coherency learning and posit that modeling sentiment coherency can provide valuable insights. Modeling sentiment coherency often presents challenges for traditional ABSC methods due to the complexity of aspect sentiment coherency. To efficiently address the aspect sentiment coherency task, we shed light on a simple yet effective approach, namely local sentiment aggregation (`LSA`). More specifically, we introduce a local sentiment aggregation paradigm powered by three unique sentiment aggregation window strategies based on various aspect-based features to guide the modeling of aspect sentiment coherency. To comprehensively evaluate 'our,' we conduct experiments for the aspect sentiment coherency extraction subtask and the tradi-

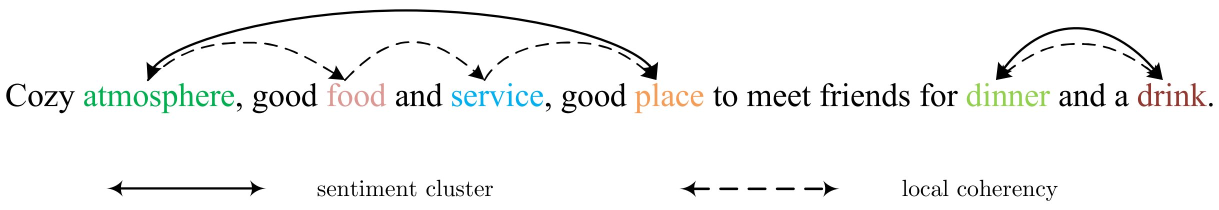

Figure 1: An example of aspect sentiment clusters and aspect sentiment coherency.

tional aspect sentiment classification subtask. Our experimental results indicate that these strategies significantly enhance sentiment coherency modeling. `LSA` achieves impressive performance in aspect sentiment coherency extraction and sentiment classification, setting new state-of-the-art results on five widely-used datasets. Therefore, our work offers a new perspective on aspect-based sentiment analysis.

In conclusion, the main contributions of our work are as follows:

- **Formulation**: We highlight the existence of sentiment coherency in ABSC and formulate the aspect sentiment coherency modeling task. Besides, we introduce a local sentiment aggregation mechanism to address this task.
- **Method**: To implement the local sentiment aggregation mechanism, we introduce three strategies for constructing sentiment aggregation windows, demonstrating the effectiveness of our model in sentiment coherency modeling. We enhance this mechanism through differential weighted sentiment aggregation, allowing for dynamic adjustment of the aggregation window construction.
- **Evaluation**: According to our extensive experimental results, `LSA` achieve impressive aspect sentiment coherency prediction results. Besides, our ensemble `LSA` model also obtains state-of-the-art aspect sentiment classification performance on five public datasets.

The code and datasets related to this work are provided in the supplementary materials.

2 Sentiment Coherency

We first introduce the concept of sentiment coherency and then formulate two sentiment coherency patterns. In the review about a restaurant in Fig. 1, the reviewer expresses positive sentiments about the atmosphere, food, and service but remains neutral about dinner and drinks. This tendency to express similar sentiments about related aspects (e.g., atmosphere, food, and service) is what we refer to as *sentiment coherency*. We calculate the number of sentiment clusters across all experimental datasets to prove this is a common phenomenon. The statistics are available in Table 1.

Our aim is to study the extraction of aspect sentiment coherency and the improvement of ABSC performance by incorporating sentiment coherency. We formulate two sentiment coherency patterns in the following sections.

2.1 Aspect Sentiment Clusters

Consider the example in Fig. 1. We notice that similar sentiments about different aspects tend to stick together, which is called *sentiment cluster*. The formulation of aspect sentiment clusters is as follows:

$$\mathcal{C} = \{C_i \mid C_i = \{a_1, a_2, \dots, a_j\}\}, \qquad (1)$$

where C_i is the i-th aspect sentiment cluster and a_j is the j-th aspect in C_i, $1 \leq j \leq m$. m is the number of identified aspects in the sentence. Aspect sentiment clustering aims at concurrently predicting all sentiment clusters based on the provided aspects. Aspect sentiment clusters can be regarded as a coarse-grained manifestation of sentiment coherency. However, directly extracting these clusters can be quite challenging. We explain the challenges in the Appendix A. In consequence, we focus on asynchronous sentiment cluster prediction based on local sentiment coherency.

2.2 Local Sentiment Coherency

We propose *"local coherency"* to simplify the modeling of aspect sentiment cluster extraction. Local coherency utilizes the aspect features to predict the sentiment iteratively. Finally, the aspects with the same sentiments are aggregated to predict sentiment clusters. There are two advantages of local sentiment coherency modeling. First, it helps us infer the sentiment about an aspect even when it isn't explicitly stated (e.g., deriving that the reviewer had a positive dining experience without saying it outright). Second, it smooths out the sentiment predictions, reducing errors caused by random noise or adversarial attacks. As a result, we can have a more accurate understanding of sentiments.

Table 1: The statistics of aspect sentiment clusters. "Cluster size" indicates the number of aspects in clusters with different sizes.

Dataset	Cluster Size					Sum
	1	2	3	4	≥ 5	
Laptop14	791	799	468	294	614	2966
Rest14	1318	1050	667	479	1214	4728
Rest15	617	406	229	163	326	1741
Rest16	836	539	314	210	462	2361
MAMS	6463	2583	1328	746	1397	12517

3 Methodology

In this section, we propose a local sentiment aggregation method for sentiment cluster prediction, which is based on the local sentiment coherency pattern. We first introduce the implementation of local sentiment aggregation, which is based on sentiment window aggregation. Then, we present the aspect feature learning method used for sentiment aggregation window construction in Section 3.2. Finally, we describe the implementation details of our model.

3.1 Local Sentiment Aggregation

To leverage local sentiment coherency, we extract the local sentiment information of each aspect and build a sentiment aggregation window (which will be clarified in Section 3.2) to aggregate coherent sentiments. In essence, the sentiment aggregation window is created by concatenating the feature representation of the aspect's local sentiment information (i.e., aspect feature in the following sections). We propose three variants, $\texttt{LSA}_P$, $\texttt{LSA}_T$, and $\texttt{LSA}_S$, to construct sentiment aggregation windows. Fig. 5 illustrates the architecture of $\texttt{LSA}_P$, while Fig. 2 presents the architecture of both $\texttt{LSA}_T$ and $\texttt{LSA}_S$. The difference between $\texttt{LSA}_T$ and $\texttt{LSA}_S$ is in the aspect feature used for local sentiment aggregation.

3.2 Aspect Feature Learning

Inspired by the existing studies, we employ the following aspect feature representations for local sentiment aggregation:

- Sentence pair-based (BERT-SPC) aspect feature (Devlin et al., 2019) (employed in $\texttt{LSA}_P$)
- Local context focus-based (LCF) aspect feature (Yang et al., 2021) (employed in $\texttt{LSA}_T$)
- Syntactical LCF-based (LCFS) based aspect feature (Phan and Ogunbona, 2020) (employed in $\texttt{LSA}_S$)

We also present an ensemble model ($\texttt{LSA}_E$) that combines the three variants of our model.

3.2.1 Sentence Pair-based Aspect Feature

A straightforward way to obtain aspect features is to utilize the BERT-SPC input format (Devlin et al., 2019), which appends the aspect to the context to learn aspect features. For example, let $\mathcal{W} = \left\{[CLS], \{w_i^c\}_{i=1}^n, [SEP], \{w_j^a\}_{j=1}^m, [SEP]\right\}$ be the BERT-SPC format input, $i \in [1, n]$ and $j \in [1, m]$, where w_i^c and w_j^a denote the token in the context and the aspect, respectively. A PLM (e.g., BERT) can learn the aspect feature because the duplicated aspects will get more attention in the self-attention mechanism (Vaswani et al., 2017). As it is shown in Fig. 5, we simply apply the sentiment aggregation to BERT-SPC-based aspect features. Note that we deploy a self-attention encoder before each linear layer to activate hidden states. We show the architecture of $\texttt{LSA}_P$ in Fig. 5.

3.2.2 Local Context-based Aspect Feature

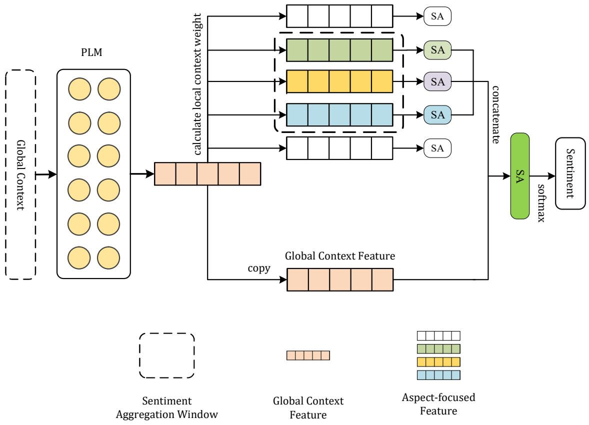

Figure 2: The local sentiment aggregation paradigm based on LCF/LCFS, denoted as $\texttt{LSA}_T$ and $\texttt{LSA}_S$.

The second implementation of our model is referred to as $\texttt{LSA}_T$. The local context-based aspect feature is derived by position-wise weighting the global context feature, where the weights are calculated using the relative distance of token-aspect pairs. Let $\mathcal{W} = \{w_1^c, w_2^c, \ldots, w_n^c\}$ be the tokens after tokenization. We calculate the position weight for token w_i^c as follows:

$$\mathbf{H}^*_{w_i^c} := \begin{cases} \mathbf{H}^c_{w_i^c} & d_{w_i^c} \leq \alpha \\ 1 - \frac{\left(d_{w_i^c} - \alpha\right)}{n} \cdot \mathbf{H}^c_{w_i^c} & d_{w_i^c} > \alpha \end{cases}, \tag{2}$$

where $\mathbf{H}^*_{w^c i}$ and $\mathbf{H}^c_{w^c i}$, $i \in [1, n]$, are the hidden states at the position of w_i^c in the aspect feature and global context feature, respectively. $d_{w_i^c}$ is the relative distance between w_i^c and the aspect. We concatenate $\mathbf{H}^*_{w_i^c}$ to obtain the aspect feature $\mathbf{H}^*$. $\alpha = 3$ is a fixed distance threshold. If $d_{w_i^c} \leq \alpha$, $\mathbf{H}^c_{w^c i}$ will be preserved; otherwise, it decays according to $d_{w_i^c}$.

In equation (2), the relative distance $d_{w_i^c}$ between w_i^c and the aspect is obtained by:

$$d_{w_i^c} := \frac{\sum_{j=1}^{m} |p_i^c - p_j^a|}{m}, \tag{3}$$

where p_i^c and p_j^a are the positions of the $w^c i$ and j-th token in the aspect. As shown in Fig. 2, we take the global context feature as a supplementary feature to learn aspect sentiments.

3.2.3 Syntactical Local Context-based Aspect Feature

The final variant of our model is $\texttt{LSA}_S$, which adopts the syntax-tree-based local context feature to construct a sentiment aggregation window. The distance between the context word w_i^c and the aspect can be calculated according to the shortest node distance between w_i^c and the aspect in the syntax tree. To leverage the syntactical information without directly modeling the syntax tree, $\texttt{LSA}_S$ calculates the average node distance between w_i^c and the aspect:

$$d_{w_i^c} = \frac{\sum_{i=j}^{m} dist(w_i^c, w_j^a)}{m}, \tag{4}$$

where $dist$ denotes the shortest distance between the node of w_i^c and the node of w_j^a in the syntax tree; the calculation of $\mathbf{H}^*_{w_i^c}$ follows $\texttt{LSA}_T$.

3.3 Sentiment Aggregation Window

The sentiment aggregation window consists of k-nearest aspect feature vectors. Given that most of the clusters are small, we only consider $k = 1$ in this study:

$$\mathbf{H}^o_{aw} := [\{\mathbf{H}^\mathbf{l}_k\}; \mathbf{H}^\mathbf{t}; \{\mathbf{H}^\mathbf{r}_k\}], \tag{5}$$

$$\mathbf{H}^o := W^o \mathbf{H}^o_{aw} + b^o, \tag{6}$$

where $\mathbf{H}^o_{aw}$ is the feature representation learned by local sentiment aggregation; ";" denotes vector concatenation. $\mathbf{H}^\mathbf{l}_k$ and $\mathbf{H}^\mathbf{r}_k$ are the k nearest left and right adjacent aspect features, respectively. $\mathbf{H}^\mathbf{t}_*$ is the targeted aspect feature. $\mathbf{H}^o_*$ is the representation learned by the sentiment aggregation window, and W^o and b^o are the trainable weights and biases.

3.3.1 Aggregation Window Padding

To handle instances with no adjacent aspects, we pad the sentiment aggregation window. Fig. 3 illustrates three padding strategies. Instead of zero

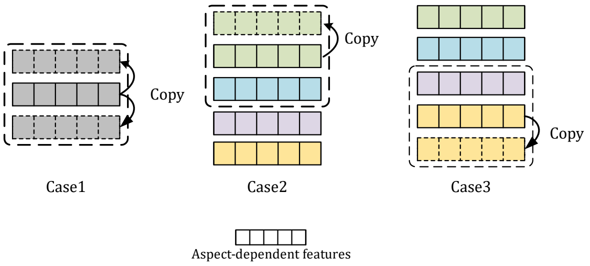

Figure 3: Window padding strategies for different situations.

vectors, we pad the window using the targeted aspect's feature to highlight the local sentiment feature of the targeted aspect and prevent the model's performance from deteriorating. Case #1 indicates a single aspect in the context, in which we triple the targeted aspect's feature to build the sentiment aggregation window. Case #2 and Case #3 duplicate the targeted aspect's feature to the left and right slots in the window, respectively.

3.3.2 Differential Weighted Aggregation

It is reasonable to assume that the importance of sentiment information from different sides may vary. Therefore, we introduce differential weighted aggregation (DWA) to control the contribution of sentiment information from the adjacent aspects on different sides. We initialize learnable η_l^* and η_r^* to 1 and optimize them using gradient descent. The differential weighted sentiment aggregation window is obtained as follows:

$$\mathbf{H}^o_{dwa} := [\eta_l^* \{\mathbf{H}^\mathbf{l}_k\}; \mathbf{H}^\mathbf{t}; \eta_r^* \{\mathbf{H}^\mathbf{r}_k\}], \tag{7}$$

where $\mathbf{H}^o_{dwa}$ is the aggregated hidden state learned by the differential weighted aggregation window.

3.4 Output Layer

For sentence pair-based sentiment aggregation, we simply apply pooling and softmax to predict the sentiment likelihood. For the local context feature-based sentiment aggregation, we adhere to the original approach of combining the global context feature and the learned feature to predict sentiment polarity as follows:

$$\mathbf{H}^{out} := W^d [\mathbf{H}^o; \mathbf{H}^c] + b^d, \tag{8}$$

where $\mathbf{H}^{out}$ is the output hidden state; $\mathbf{H}^o$ and $\mathbf{H}^c$ are the features extracted by a PLM (e.g., DeBERTa). We use the feature of the first token (also known as the head pooling) to classify sentiments:

$$\hat{y} := \frac{\exp(\mathbf{h}^{head})}{\sum_1^{\tilde{C}} \exp(\mathbf{h}^{head})}, \tag{9}$$

where $\mathbf{h}^{head}$ is the head-pooled feature; $\tilde{C}$ is the number of polarity categories. $W^d \in \mathbb{R}^{1\times\tilde{C}}$, $b^d \in \mathbb{R}^{\tilde{C}}$ are the trainable weights and biases. $\hat{y}$ is the predicted sentiment polarity.

3.5 Training Details

The variants of our model based on different PLMs are denoted as LSA-BERT, LSA-RoBERTa, LSA-DeBERTa, etc. LSA−X represents our model based on the large version of PLM.

We train our model using the AdamW optimizer with the cross-entropy loss function:

$$\mathcal{L} = -\sum_1^{\tilde{C}} \widehat{y}_i \log y_i + \lambda ||\Theta||_2 + \lambda^* ||\eta_l^*, \eta_r^*||_2, \tag{10}$$

where λ is the L_2 regularization parameter; Θ is the parameter set of the model. As we employ gradient-based optimization for η_l^* and η_r^*, we also apply a L_2 regularization with λ^* for η_l^* and η_r^*.

4 Experiments

In this section, we introduce the settings of our experiments and report the experimental results. We report all implementation details in the appendix, e.g., hyperparameter settings (Appendix B.2), baseline introduction (Appendix B.3) and additional experiments, etc.

4.1 Datasets

To evaluate the efficacy of the local sentiment aggregation, we conducted experiments on five popular ABSC datasets [1]: Laptop14, Rest14, Rest15 and Rest16 datasets, and MAMS dataset (Jiang et al., 2019), respectively. The statistics of these datasets are shown in Table 2.

4.2 Baselines

Please refer to Appendix B.3 for the introduction of baselines.

[1] We evaluate LSA on the Twitter (Dong et al., 2014) dataset and report the experimental results in Section C.4. The processed datasets are available with the code in supplementary materials.

Table 2: The statistics of all datasets used in our experiments. Note that in our experiments, only the MAMS dataset has a validation set.

Datasets	Positive		Negative		Neutral	
	Train	Test	Train	Test	Train	Test
Laptop14	994	341	870	128	464	169
Rest14	2164	728	807	196	637	196
Rest15	909	326	256	180	36	34
Rest16	1240	468	437	117	69	30
MAMS	3379	400	2763	329	5039	607

4.3 Main Results

We report sentiment coherency modeling performance and sentiment classification performance in this section.

Table 3: The exact match score of sentiment cluster prediction on five public datasets The best results are highlighted in **bold** font.

Model	Laptop14	Rest14	Rest15	Rest16	MAMS
	EM	EM	EM	EM	EM
BERT	75.08	78.75	80.00	87.60	79.26
DeBERTa	79.61	83.88	84.05	89.72	81.16
LSA_P-BERT	78.14	82.24	82.76	88.96	82.35
LSA_T-BERT	78.06	82.96	82.66	90.02	82.46
LSA_S-BERT	78.63	83.09	83.30	88.75	82.73
LSA_E-BERT	78.94	83.62	83.40	89.96	84.03
LSA_P-DeBERTa	82.55	86.39	86.93	92.14	82.83
LSA_T-DeBERTa	81.96	86.26	87.03	91.72	83.38
LSA_S-DeBERTa	82.94	85.90	87.13	91.87	83.92
LSA_E-DeBERTa	**83.73**	**86.53**	**87.91**	**92.57**	**84.12**

4.3.1 Cluster Prediction Performance

We utilize LSA to classify aspect sentiments and aggregate the sentiment clusters. The cluster prediction performance in Table 3 shows that our models consistently outperform the baseline models on all datasets. The performance of LSA is dependent on the base model. It is observed that the sentiment clusters predicted by LSA are very close to the ground truth, which demonstrates the effectiveness of our models in modeling sentiment coherency. The small clusters (e.g., clusters containing 1 or 2 aspects) are more easy to predict, while the large clusters (e.g., ≥ 3) are more difficult to predict.

4.3.2 Sentiment classification performance

When it comes to sentiment classification performance, the results in Table 4 clearly demonstrate the superiority of our models over significant baselines, particularly in the case of the LSA_E model. The experimental results are as expected and show the proficiency of LSA.

One of the primary concerns associated with LSA is its occasional inability to outperform certain baselines based on the BERT model. We attribute this observation to two main reasons.

Table 4: The traditional aspect sentiment classification performance on five public datasets, and the best results are heightened in **bold** font. † indicates the results are the best performance in multiple runs, while other methods report the average performance. ‡ indicates the experimental results of the models implemented by us.

Model		Laptop14		Restaurant14		Restaurant15		Restaurant16		MAMS	
		Acc	F1	Acc	F1	Acc	F1	Acc	F1	Acc	F1
SK-GCN-BERT (Zhou et al., 2020)	Baselines	79.00	75.57	83.48	75.19	83.20	66.78	87.19	72.02	—	—
SDGCN-BERT (Zhao et al., 2020)		81.35	78.34	83.57	76.47	—	—	—	—	—	—
DGEDT-BERT (Tang et al., 2020)		79.80	75.60	86.30	80.00	84.00	71.00	91.90	79.00	—	—
DualGCN-BERT (Li et al., 2021a)		81.80	78.10	87.13	81.16	—	—	—	—	—	—
TF-BERT (Zhang et al., 2023)		81.80	78.46	87.09	81.15	—	—	—	—	—	—
dotGCN-BERT (Chen et al., 2022)		81.03	78.10	86.16	80.49	—	—	—	—	—	—
SSEGCN-BERT (Zhang et al., 2022)		81.01	77.96	87.31	81.09	—	—	—	—	—	—
TGCN-BERT (Li et al., 2021a)		80.88	77.03	86.16	79.95	83.38	82.77	86.00	72.81	—	—
ASGCN-RoBERTa Dai et al. (2021)		83.33	80.32	86.87	80.59	—	—	—	—	—	—
RGAT-RoBERTa Dai et al. (2021)		83.33	79.95	87.52	81.29	—	—	—	—	—	—
PWCN-RoBERTa Dai et al. (2021)		84.01	81.08	87.35	80.85	—	—	—	—	—	—
SARL-RoBERTa† (Wang et al., 2021)		85.42	82.97	88.21	82.44	88.19	73.83	94.62	81.92	—	—
RoBERTa (Liu et al., 2019)‡		82.76(0.63)	79.73(0.77)	87.77(1.61)	82.10(2.01)	78.06(0.55)	62.41(0.89)	93.01(0.19)	80.88(0.27)	83.83(0.49)	83.29(0.50)
DeBERTa (He et al., 2021)‡		82.76(0.31)	79.45(0.60)	88.66(0.35)	83.06(0.29)	87.50(0.28)	73.76(0.36)	86.57(0.78)	73.59(0.95)	83.06(1.24)	82.52(1.25)
SARL-DeBERTa‡ (Wang et al., 2021)		83.32(0.42)	79.95(0.51)	86.69(0.27)	78.91(0.33)	86.53(0.19)	69.73(0.28)	93.31(0.19)	80.13(0.28)	82.03(0.57)	81.84(0.28)
LSA_P-BERT	LSA	81.35(0.63)	77.79(0.48)	87.23(0.22)	81.06(0.67)	84.07(0.78)	70.62(0.68)	91.74(0.32)	78.25(0.88)	83.13(0.30)	82.53(0.44)
LSA_T-BERT		81.35(0.39)	78.43(0.52)	87.32(0.22)	81.86(0.20)	84.93(0.59)	73.01(0.79)	91.42(0.45)	77.50(0.86)	83.51(0.26)	82.90(0.28)
LSA_S-BERT		81.03(0.31)	77.45(0.37)	87.41(0.40)	81.52(0.49)	84.22(1.03)	71.98(0.85)	91.58(0.54)	77.54(0.71)	83.23(0.56)	82.68(0.52)
LSA_S-BERT		81.03(0.31)	77.45(0.37)	87.41(0.40)	81.52(0.49)	85.56(0.41)	73.79(0.57)	92.20(0.63)	78.49(0.65)	83.23(0.56)	82.68(0.52)
LSA_P-RoBERTa		83.39(0.35)	80.47(0.44)	88.04(0.62)	82.96(0.48)	87.01(0.18)	73.71(0.31)	90.31(0.94)	76.17(1.48)	83.37(0.31)	83.78(0.29)
LSA_T-RoBERTa		83.44(0.56)	80.47(0.71)	88.30(0.37)	83.09(0.45)	86.64(0.57)	72.24(0.79)	94.22(0.71)	83.41(1.45)	83.31(0.41)	84.60(0.22)
LSA_S-RoBERTa		83.23(0.44)	80.30(0.68)	88.48(0.52)	83.81(0.62)	88.31(0.47)	76.23(0.81)	93.65(0.89)	81.82(1.71)	83.58(0.39)	83.78(0.24)
LSA_E-RoBERTa		84.12(0.27)	80.90(0.51)	89.11(0.38)	83.98(0.69)	88.39(0.53)	76.19(0.68)	94.15(0.64)	82.18(1.38)	85.48(0.29)	85.02(0.17)
LSA_P-DeBERTa		84.33(0.55)	81.46(0.77)	89.91(0.09)	84.90(0.45)	89.05(0.28)	77.14(0.37)	93.49(0.43)	81.44(0.53)	83.91(0.31)	83.31(0.21)
LSA_T-DeBERTa		84.80(0.39)	82.00(0.43)	89.91(0.40)	85.05(0.85)	89.61(0.72)	79.17(0.12)	93.65(0.39)	81.53(0.51)	84.28(0.32)	83.70(0.47)
LSA_S-DeBERTa		84.17(0.08)	81.23(0.27)	89.64(0.66)	84.53(0.79)	89.42(0.38)	77.29(0.62)	94.14(0.11)	81.61(0.81)	83.61(0.30)	83.07(0.28)
LSA_E-DeBERTa		84.80(0.31)	82.09(0.31)	91.43(0.28)	86.85(0.19)	89.47(0.59)	77.84(0.40)	94.47(0.37)	82.39(0.27)	85.85(0.18)	85.29(0.37)
LSA_P-X-DeBERTa		86.00(0.07)	83.10(0.30)	90.27(0.61)	85.51(0.48)	89.98(0.11)	78.26(0.98)	95.11(0.69)	84.68(0.21)	82.78(0.96)	81.99(0.86)
LSA_T-X-DeBERTa		86.31(0.20)	83.93(0.27)	90.86(0.18)	86.26(0.22)*b*	91.09(0.22)	81.22(0.34)	94.71(0.56)	84.34(0.38)	84.21(0.42)	83.72(0.46)
LSA_S-X-DeBERTa		86.21(0.52)	83.97(0.64)	90.33(0.37)	85.55(0.46)	90.63(0.17)	80.24(0.33)	94.54(0.84)	83.50(0.73)	84.68(0.67)	84.12(0.64)
LSA_E-X-DeBERTa		**86.46(0.38)**	**84.41(0.39)**	**90.98(0.28)**	**87.02(0.42)**	**91.85(0.27)**	**81.29(0.51)**	**95.61(0.64)**	**84.87(0.71)**	**86.38(0.29)**	**85.97(0.18)**

Firstly, LSA is a quite simple mechanism and relies on relatively basic aspect features to construct sentiment aggregation windows, which may not be as competitive as state-of-the-art methods that employ more complex features. Secondly, the current sentiment aggregation window, although intuitive, may not be perfect and could potentially lead to the loss of some sentiment information. Nevertheless, the performance of the three LSA variants may not consistently surpass some baselines, our models offer notable advantages in terms of efficiency and ease of integration with existing models. With the improvement in the base model's performance (e.g., DeBERTa, DeBERTa-Large), LSA achieves impressive results across all datasets. Furthermore, it's worth noting that methods such as ASGCN-RoBERTa, RGAT-RoBERTa, and PWCN-RoBERTa, while showing promising improvements, come at the cost of significantly higher resource requirements compared to other models.

In summary, LSA presents a compelling choice for a trade-off between performance and resource efficiency with the potential to be integrated into existing models with minimal effort.

4.4 Practice in Adversarial Defense

Recent works have highlighted the threat of textual adversarial attacks (Xing et al., 2020) as critical threats. In this section, we embark on a pioneering exploration of LSA's capabilities, focusing on its ability to defend against adversarial attacks in ABSC. To evaluate the robustness of LSA in the face of these attacks, we employ existing adversarial attack datasets, specifically Lap14-ARTS and Rest14-ARTS[2].

Table 5: Performance comparison of different models for adversarial defense on the Lap14-ARTS and Rest14-ARTS datasets. The adversarial datasets are from Xing et al. (2020).

Model	Lap14-ARTS		Rest14-ARTS	
	Acc	F1	Acc	F1
BERT	63.98	56.11	72.01	65.62
DeBERTa	67.71	65.60	74.97	66.48
LSA_P-BERT	72.31	68.94	78.06	70.23
LSA_T-BERT	72.12	68.05	77.57	70.72
LSA_S-BERT	70.88	65.98	77.99	71.01
LSA_E-BERT	**74.32**	**69.57**	**78.41**	**72.04**
LSA_P-DeBERTa	73.34	68.46	81.19	72.54
LSA_T-DeBERTa	73.58	69.28	80.31	71.37
LSA_S-DeBERTa	72.31	67.03	79.13	71.82
LSA_E-DeBERTa	**74.47**	**69.79**	**81.55**	**72.95**

The results presented in Table 5 serve as a testament to the superior performance of our models when compared to the baseline models, i.e., BERT and DeBERTa. Notably, when considering the DeBERTa-based models, LSA_P-DeBERTa, LSA_T-DeBERTa, and LSA_S-DeBERTa consistently outperform the baselines, underscoring the robustness of LSA in defend against adversarial attack.

[2]We will provide access to all our experiments through our code.

4.5 Ablation Study

In this section, we study how gradient-based aggregation window optimization influences LSA. We begin by presenting the trajectory of η_l^* and η_r^* during the training process, as depicted in Fig. 4, which illustrates how LSA dynamically constructs the optimal window. This observation suggests that the model initially prioritizes the side aspects during early training stages, gradually shifting focus towards the central aspects. To further investigate the impact of gradient-based aggregation window optimization, we conduct a comparative analysis by evaluating LSA's performance with and two ablated models without DWA. Specifically, we assess the model's performance when employing fixed static weights η_l and η_r to create sentiment aggregation windows, as opposed to the DWA. The experimental results provided in Fig. 6 demonstrate a consistent performance drop when DWA is omitted. In most scenarios, we observe a modest yet notable improvement of approximately 0.2% to 0.5% when DWA is incorporated into our model. We also present the experimental results for an ablated version of LSA featuring a simplified sentiment aggregation window in Table 9. This comparison underscores the superior performance of LSA with DWA over its simplified counterpart. Consequently, we can conclude that gradient-based aggregation window optimization proves effective in facilitating implicit sentiment learning.

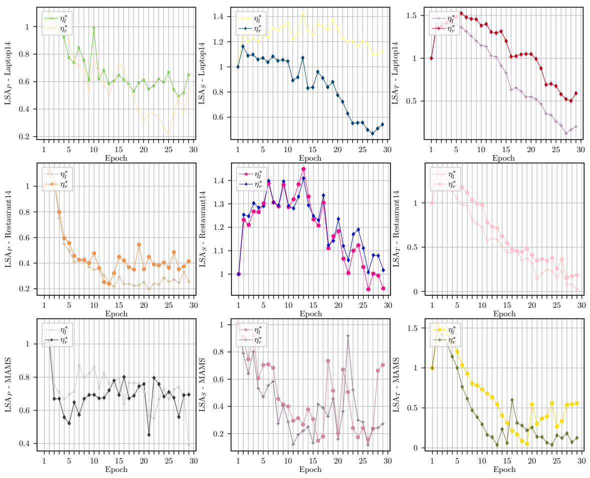

Figure 4: Trajectory visualization of learnable weights in gradient-based sentiment aggregation window optimization.

4.6 Case Study

In this section, we delve into a case study to validate the capability of our model in learning local sentiment coherency. We present a series of examples in Table 6, which showcase instances where LSA excels in identifying aspect sentiment coherency.

Table 6: The examples for aspect sentiment coherency found by LSA. The target aspects are denoted in **bold** and the underlined words indicates the aspects with coherent sentiments. "Pos", "Neg" and "Neu" represent positive, negative and neutral, respectively.

No.	Domain	Examples	Model	Prediction
1	Restaurant	Not only was the *food* outstanding, but also the **coffee** and **juice**!	$\mathtt{LSA}_{\mathcal{P}}$-BERT	Pos(Pos) ✓, Pos(Pos) ✓
		Not only was the *food* terrible, but also the **coffee** and **juice**!	$\mathtt{LSA}_{\mathcal{P}}$-BERT	Neg(Neg) ✓, Neu(Neg) ✗
2	Restaurant	The *servers* always surprise us with a different **starter**.	$\mathtt{LSA}_{\mathcal{S}}$-BERT	Pos(Pos) ✓
		The *servers* always temporize us with a different **starter**.	$\mathtt{LSA}_{\mathcal{S}}$-BERT	Neg(Neg) ✓
3	TV	The speakers of this TV is great! Just like its **screen**.	$\mathtt{LSA}_{\mathcal{T}}$-DeBERTa	Pos(Pos) ✓
		The speakers of this TV sucks! Just like its **screen**.	$\mathtt{LSA}_{\mathcal{T}}$-DeBERTa	Neg(Neg) ✓
4	Camera	If you are worried about **usability**, think about the *quality* !	DeBERTa	Neu(Pos) ✗
		If you are worried about **usability**, think about it good *quality* !	DeBERTa	Pos(Pos) ✓

These examples offer compelling evidence of the effectiveness of our model, as compared to a baseline model (DeBERTa). For instance, in example #4, the DeBERTa model produces two inference errors in recognizing coherent sentiments, while all our model variants based on the DeBERTa model yield correct results. Furthermore, $\mathtt{LSA}_{\mathcal{P}}$, $\mathtt{LSA}_{\mathcal{T}}$, and $\mathtt{LSA}_{\mathcal{S}}$models demonstrate remarkable robustness in handling perturbed examples that involve local sentiment coherency. While it is challenging to present a comprehensive list of sentiment cluster prediction examples, the consistent observations obtained in these experiments align with those in Table 6. Based on these experimental results, we confidently assert the model's proficiency in learning sentiment coherency within ABSC.

5 Discussions

5.1 How can LSA help to existing methods?

The primary function of LSA lies in aggregating aspect features based on local sentiment coherency. Thanks to its straightforward implementation, integrating LSA into existing models is a seamless process. In practice, once aspect features have been extracted using any existing methods, LSA can be effortlessly applied to extract aspect sentiment clusters, enhancing the overall performance of aspect sentiment classification.

A simple yet effective way to incorporate LSA into existing models involves removing their output layer and passing the learned feature representations of adjacent aspects to LSA. Subsequently, LSA can construct the sentiment aggregation window and derive the weights for each aspect feature using the Differential Weighted Aggregation (DWA) method.

5.2 How does LSA works on adverse sentiment aggregation?

In this section, we justify why LSA works for adjacent but inconsistent sentiment. It is intuitively that not all aspect sentiments in adjacent positions are similar but sometimes be opposite. However, LSA learns to discriminate whether they share similar sentiments based on the training data. If no local sentiment coherency is detected, LSA learns a weight close to 0 to the feature of adjacent aspects in the DWA.

We have conducted experiments on a sub-dataset extracted from the MAMS dataset that only includes both marginal aspects in clusters, denoted as Margin dataset. We evaluate the sentiment prediction accuracy of aspects near inconsistent sentiment clusters. The results are available in Table 7, and the performance of classifying margin aspects is still comparable to global performance in Table 4, indicating that differentiated weighting for LSA effectively mitigates the challenge of adverse sentiment aggregation.

Table 7: The performance of sentiment predictions for margin aspects in various models on the MAMS dataset.

Model	Margin		MAMS	
	Acc	F1	Acc	F1
LSA_P-DeBERTa	83.49	82.71	**83.91**	**83.31**
LSA_T-DeBERTa	82.58	81.79	**84.28**	**83.70**
LSA_S-DeBERTa	**83.87**	**83.11**	83.61	83.07

6 Related Works

The related works in this field can be broadly divided into three categories: sentiment dependency-based methods, sentiment coherency modeling, and implicit sentiment learning.

Although sentiment coherency is prevalent in ABSC, it has received limited attention in recent years. However, the progress of sentiment dependency-based methods, such as the work by Zhang et al. (2019); Zhou et al. (2020); Tian et al. (2021); Li et al. (2021a); Dai et al. (2021), has contributed to the improvement of coherent sentiment learning. These studies explored the effectiveness of syntax information in ABSC, which mitigates issues related to sentiment coherency extraction.

For refining syntax structure quality in sentiment dependency learning, Tian et al. (2021) employ type-aware GCN to distinguish different relations in the graph, achieving promising results. Similarly, Li et al. (2021a) propose SynGCN and SemGCN for different dependency information. TGCN model alleviates dependency parsing errors and shows significant improvement compared to previous GCN-based models. Despite the aforementioned advances, transferring the new techniques proposed in these studies is not straightforward. Dai et al. (2021) propose employing the pre-trained RoBERTa model to induce trees for ABSC, effectively solving the node alignment problem. However, the efficiency of inducing trees needs improvement.

Compared to coarse-grained implicit sentiment research (de Kauter et al., 2015; Zhou et al., 2021; Liao et al., 2022; Zhuang et al., 2022), the aspect's implicit sentiment learning in ABSC remains challenging. LSA leverages coherency to aggregate implicit sentiments efficiently. Some researchers have formulated tasks aimed at modeling implicit sentiments and opinions. For instance, Cai et al. (2021) proposed a quadruple extraction task (aspect, category, opinion, and sentiment), while Murtadha et al. (2022) proposed a unified framework that crafts auxiliary sentences to aid implicit aspect extraction and sentiment analysis. In contrast to these works, LSA sidesteps the efficiency bottleneck of syntax modeling by eliminating structure information and proves to be adaptable to existing methods as it is a transferable paradigm independent of base models. Li et al. (2021b) presents a supervised contrastive pre-training mechanism to align the representation of implicit sentiment and explicit sentiment. However, it relies on fine-tuning a large-scale sentiment-annotated corpus from in-domain language resources, which may be resource-intensive and inefficient.

7 Conclusion

Aspect sentiment coherency has been overlooked in existing studies. We introduced the concept of LSA, a novel approach that brings the nuance of local sentiment coherency into the foreground of ABSC. This approach achieves state-of-the-art performance when combined with various base models. Furthermore, we also introduce a practice of LSA in the realm of adversarial defense. We hope that our work will inspire further research into sentiment coherency modeling in the future.

8 Limitation

Although LSA achieves impressive performance for multiple-aspects situations, e.g., SemEval-2014 datasets. However, while being applied in mono aspect situations, such as the Twitter dataset, LSA degenerates to be equivalent to a prototype model, e.g., the local context focus model.

Another limitation is that LSA is a quite simple mechanism and relies on relatively basic aspect features to construct sentiment aggregation windows, which may not be as competitive as state-of-the-art methods that employ more complex features. Besides, the current sentiment aggregation window is intuitive but may not be perfect and could potentially lead to the loss of some sentiment information. In the future, we will explore more advanced sentiment aggregation windows to improve the performance of LSA.

References

Hongjie Cai, Rui Xia, and Jianfei Yu. 2021. Aspect-category-opinion-sentiment quadruple extraction with implicit aspects and opinions. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, ACL/IJCNLP 2021, (Volume 1: Long Papers), Virtual Event, August 1-6, 2021*, pages 340–350. Association for Computational Linguistics.

Jiahao Cao, Rui Liu, Huailiang Peng, Lei Jiang, and Xu Bai. 2022. Aspect is not you need: No-aspect differential sentiment framework for aspect-based sentiment analysis. In *NAACL-HLT*, pages 1599–1609. Association for Computational Linguistics.

Chenhua Chen, Zhiyang Teng, Zhongqing Wang, and Yue Zhang. 2022. Discrete opinion tree induction for aspect-based sentiment analysis. In *ACL (1)*, pages 2051–2064. Association for Computational Linguistics.

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. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2021, Online, June 6-11, 2021*, pages 1816–1829. Association for Computational Linguistics.

Marjan Van de Kauter, Diane Breesch, and Véronique Hoste. 2015. Fine-grained analysis of explicit and implicit sentiment in financial news articles. *Expert Syst. Appl.*, 42(11):4999–5010.

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, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 4171–4186. Association for Computational Linguistics.

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, ACL 2014, June 22-27, 2014, Baltimore, MD, USA, Volume 2: Short Papers*, pages 49–54. The Association for Computer Linguistics.

Feifan Fan, Yansong Feng, and Dongyan Zhao. 2018. Multi-grained attention network for aspect-level sentiment classification. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*, pages 3433–3442. Association for Computational Linguistics.

Pengcheng He, Jianfeng Gao, and Weizhu Chen. 2021. Debertav3: Improving deberta using electra-style pre-training with gradient-disentangled embedding sharing. *CoRR*, abs/2111.09543.

Binxuan Huang and Kathleen M. Carley. 2019. Syntax-aware aspect level sentiment classification with graph attention networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019*, pages 5468–5476. Association for Computational Linguistics.

Qingnan Jiang, Lei Chen, Ruifeng Xu, Xiang Ao, and Min Yang. 2019. A challenge dataset and effective models for aspect-based sentiment analysis. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019*, pages 6279–6284. Association for Computational Linguistics.

Ruifan Li, Hao Chen, Fangxiang Feng, Zhanyu Ma, Xiaojie Wang, and Eduard H. Hovy. 2021a. Dual graph convolutional networks for aspect-based sentiment analysis. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, ACL/IJCNLP 2021, (Volume 1: Long Papers), Virtual Event, August 1-6, 2021*, pages 6319–6329. Association for Computational Linguistics.

Zhengyan Li, Yicheng Zou, Chong Zhang, Qi Zhang, and Zhongyu Wei. 2021b. Learning implicit sentiment in aspect-based sentiment analysis with supervised contrastive pre-training. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 7-11 November, 2021*, pages 246–256. Association for Computational Linguistics.

Jian Liao, Min Wang, Xin Chen, Suge Wang, and Kai Zhang. 2022. Dynamic commonsense knowledge fused method for chinese implicit sentiment analysis. *Inf. Process. Manag.*, 59(3):102934.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized BERT pretraining approach. *CoRR*, abs/1907.11692.

Dehong Ma, Sujian Li, Xiaodong Zhang, and Houfeng Wang. 2017. Interactive attention networks for aspect-level sentiment classification. In *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI 2017, Melbourne, Australia, August 19-25, 2017*, pages 4068–4074. ijcai.org.

Ahmed Murtadha, Shengfeng Pan, Bo Wen, Jianlin Su, Wenze Zhang, and Yunfeng Liu. 2022. BERT-ASC: auxiliary-sentence construction for implicit aspect learning in sentiment analysis. *CoRR*, abs/2203.11702.

Minh Hieu Phan and Philip O. Ogunbona. 2020. Modelling context and syntactical features for aspect-based sentiment analysis. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 3211–3220. Association for Computational Linguistics.

Maria Pontiki, Dimitris Galanis, Haris Papageorgiou, Ion Androutsopoulos, Suresh Manandhar, Mohammad Al-Smadi, Mahmoud Al-Ayyoub, Yanyan Zhao, Bing Qin, Orphée De Clercq, Véronique Hoste, Marianna Apidianaki, Xavier Tannier, Natalia V. Loukachevitch, Evgeniy V. Kotelnikov, Núria Bel, Salud María Jiménez Zafra, and Gülsen Eryigit. 2016. Semeval-2016 task 5: Aspect based sentiment analysis. In *Proceedings of the 10th International Workshop on Semantic Evaluation, SemEval@NAACL-HLT 2016, San Diego, CA, USA, June 16-17, 2016*, pages 19–30. The Association for Computer Linguistics.

Maria Pontiki, Dimitris Galanis, Haris Papageorgiou, Suresh Manandhar, and Ion Androutsopoulos. 2015. Semeval-2015 task 12: Aspect based sentiment analysis. In *Proceedings of the 9th International Workshop on Semantic Evaluation, SemEval@NAACL-HLT 2015, Denver, Colorado, USA, June 4-5, 2015*, pages 486–495. The Association for Computer Linguistics.

Maria Pontiki, Dimitris Galanis, John Pavlopoulos, Harris Papageorgiou, Ion Androutsopoulos, and Suresh Manandhar. 2014. Semeval-2014 task 4: Aspect based sentiment analysis. In *Proceedings of the 8th International Workshop on Semantic Evaluation, SemEval@COLING 2014, Dublin, Ireland, August 23-24, 2014*, pages 27–35. The Association for Computer Linguistics.

Hao Tang, Donghong Ji, Chenliang Li, and Qiji Zhou. 2020. Dependency graph enhanced dual-transformer structure for aspect-based sentiment classification. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 6578–6588. Association for Computational Linguistics.

Yuanhe Tian, Guimin Chen, and Yan Song. 2021. Aspect-based sentiment analysis with type-aware graph convolutional networks and layer ensemble. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2021, Online, June 6-11, 2021*, pages 2910–2922. Association for Computational Linguistics.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 5998–6008.

Bo Wang, Tao Shen, Guodong Long, Tianyi Zhou, and Yi Chang. 2021. Eliminating sentiment bias for aspect-level sentiment classification with unsupervised opinion extraction. In *Findings of the Association for Computational Linguistics: EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 16-20 November, 2021*, pages 3002–3012. Association for Computational Linguistics.

Xiaoyu Xing, Zhijing Jin, Di Jin, Bingning Wang, Qi Zhang, and Xuanjing Huang. 2020. Tasty burgers, soggy fries: Probing aspect robustness in aspect-based sentiment analysis. In *EMNLP (1)*, pages 3594–3605. Association for Computational Linguistics.

Heng Yang, Biqing Zeng, Jianhao Yang, Youwei Song, and Ruyang Xu. 2021. A multi-task learning model for chinese-oriented aspect polarity classification and aspect term extraction. *Neurocomputing*, 419:344–356.

Chen Zhang, Qiuchi Li, and Dawei Song. 2019. Aspect-based sentiment classification with aspect-specific graph convolutional networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019*, pages 4567–4577. Association for Computational Linguistics.

Mao Zhang, Yongxin Zhu, Zhen Liu, Zhimin Bao, Yunfei Wu, Xing Sun, and Linli Xu. 2023. Span-level aspect-based sentiment analysis via table filling. In *ACL (1)*, pages 9273–9284. Association for Computational Linguistics.

Zheng Zhang, Zili Zhou, and Yanna Wang. 2022. SSEGCN: syntactic and semantic enhanced graph convolutional network for aspect-based sentiment analysis. In *NAACL-HLT*, pages 4916–4925. Association for Computational Linguistics.

Pinlong Zhao, Linlin Hou, and Ou Wu. 2020. Modeling sentiment dependencies with graph convolutional networks for aspect-level sentiment classification. *Knowl. Based Syst.*, 193:105443.

Deyu Zhou, Jianan Wang, Linhai Zhang, and Yulan He. 2021. Implicit sentiment analysis with event-centered text representation. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 7-11 November, 2021*, pages 6884–6893. Association for Computational Linguistics.

Jie Zhou, Jimmy Xiangji Huang, Qinmin Vivian Hu, and Liang He. 2020. SK-GCN: modeling syntax and knowledge via graph convolutional network for aspect-level sentiment classification. *Knowl. Based Syst.*, 205:106292.

Yin Zhuang, Zhen Liu, Tingting Liu, Chih-Chieh Hung, and Yan-Jie Chai. 2022. Implicit sentiment analysis based on multi-feature neural network model. *Soft Comput.*, 26(2):635–644.

A Challenges of Aspect Sentiment Cluster Extraction

The challenges of concurrent aspect sentiment cluster extraction can be summarized in the following three aspects:

- **Data Annotation:** Currently, there is no existing aspect cluster dataset in the literature since addressing sentiment coherence is a novel topic. Re-annotating cluster data and labels presents a significant challenge, and modeling these clusters is notably more complex when contrasted with local sentiment coherence aggregation.

- **Data Insufficiency:** Even after completing the data re-annotation process, the clusters within the datasets might still be insufficient for effectively training the model.

- **Modeling Difficulty:** Cluster mining is a hard task compared to text classification, but it is worth studying in the near future.

B Implementation Details

B.1 Model Architecture

We show the brief architecture of $\texttt{LSA}_P$ (based on the BERT-SPC input format) in Fig. 5. The input of $\texttt{LSA}_P$ is the same as BERT-SPC, which is a sequence of tokens with the aspect marked by the `[ASP]` token.

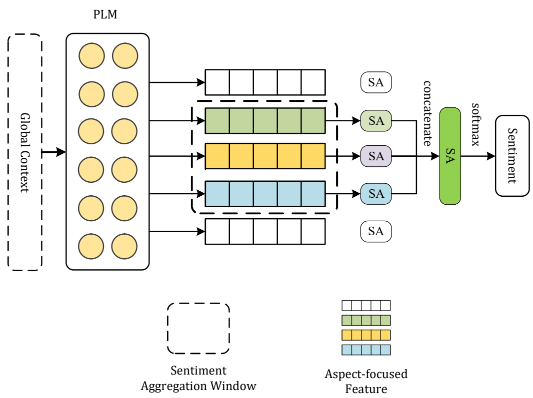

Figure 5: The local sentiment aggregation paradigm based on BERT-SPC, denoted as $\texttt{LSA}_P$. "SA" indicates the self-attention encoder.

B.2 Hyperparameter Settings

We fine-tune `LSA` using the following hyperparameters which are obtained by grid searching.

- We set $k = 1$ in sentiment aggregation window construction.
- The learning rate for pre-trained models (e.g., `BERT` and `DeBERTa`) is 2×10^{-5}.
- The learning rates for η_l^* and η_r^* are both 0.01.
- The batch size and maximum text modeling length are 16 and 80, respectively.
- The L_2 regularization parameters λ and λ_* are both 10^{-5}.

We conduct experiments based on multiple PLMs. We implement our model based on the transformers: `https://github.com/huggingface/transformers`.

B.3 Compared Models

In our comparative analysis, we evaluate the performance of `LSA` in relation to several state-of-the-art ABSC models, many of which are syntax-based methods. These models include `SK-GCN-BERT`(Zhou et al., 2020), which utilizes graph convolutional networks (GCN) to incorporate syntax and commonsense information for sentiment learning. `DGEDT-BERT`(Tang et al., 2020) is a dual-transformer-based network enhanced by a dependency graph, while `SDGCN-BERT`(Zhao et al., 2020) is a GCN-based model designed to capture sentiment dependencies between aspects. `Dual-GCN`(Li et al., 2021a) is an innovative GCN-based model that enhances the learning of syntax and semantic features.

Additionally, we include models improved by Dai et al. (2021), such as `RGAT-RoBERTa`, `PWCN-RoBERTa`, and `ASGCN-RoBERTa`, which leverage `RoBERTa` to induce syntax trees that align with `RoBERTa`'s tokenization strategy. `TGCN-BERT`(Tian et al., 2021) introduces a type-aware GCN that uses an attention mechanism to measure the importance of each edge in the syntax structure graph. `SARL-RoBERTa`(Wang et al., 2021) employs adversarial training to mitigate sentiment bias and align aspects with opinion words using span-based dependency. Finally, `dotGCN-BERT`(Chen et al., 2022), `SSEGCN-BERT`(Zhang et al., 2022), and `TGCN-BERT` (Li et al., 2021a) are also included in our comparison. These models represent the current landscape of ABSC research, allowing us to assess the effectiveness of `LSA` against well-established approaches.

We do not compare with Cao et al. (2022) because we fail to find the source code of their model.

C Additional Experimental Results

C.1 Resource Occupation of `LSA`

The experiments are based on RTX2080 GPU, AMD R5-3600 CPU with PyTorch 1.9.0. The original size of the `Laptop14` and `Restaurant14` datasets are $336kb$ and $492kb$, respectively.

Table 8: The resources occupation of state-of-the-art ABSC models. "`Proc.T.`" and "`Add.S.`" indicate the dataset pre-processing time (sec.) and additional storage occupation (MB), respectively. "*" represents non-syntax tree based models, and "†" indicates our models.

Model	Laptop14		Restaurant14	
	Proc.T.	Add.S.	Proc.T.	Add.S.
BERT-BASE *	1.62	0	3.17	0
LCF-BERT *	2.89	0	3.81	0
ASGCN-BERT	13.29	0.01	0.02	9.4
RGAT-BERT	35.4k	157.4	48.6k	188
LSA_T-BERT*†	3.16	0	4.32	0
LSA_S-BERT*†	20.56	0	30.23	0
LSA_P-BERT*†	0.20	0	0.32	0

C.2 Experiment of Static Weighted Sentiment Aggregation

Besides the dynamic sentiment window differential weighting, we also try static weight to control the contribution of adjacent aspects' sentiment information. We first initialize η_l, $\eta \in [0,1]$), for the left-adjacent aspects, while $\eta_r = 1 - \eta_l$. In this case, a greater η_l means more importance of the left-adjacent aspect's feature and vice versa. However, it is difficult to search for the optimal static weights for many scenarios via gird search. We even found that the performance trajectory is non-convex while $\eta_l \in [0,1]$, indicating the η_l on a dataset will be difficult to reuse on another dataset. Fig. 6 shows the performance curve of `LSA` based on `DeBERTa` under different η_l.

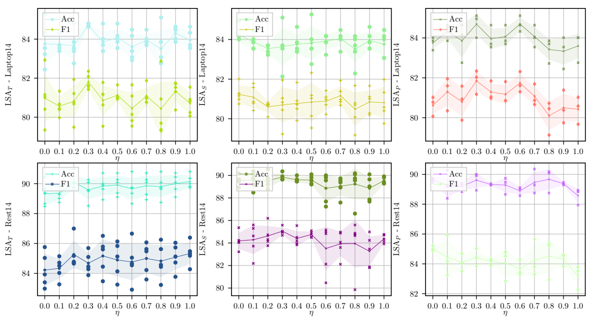

Figure 6: Visualization of performance under static differential weighting.

In other words, static differential weighting is inefficient and unstable. We recommend applying an automatic weights search to find a better construction strategy for the sentiment window.

C.3 Experiment of Simplified Sentiment Aggregation Window

To investigate the necessity of bidirectional aggregation, we assess the effectiveness of the streamlined aggregation window. We simply concatenate the left or right adjacent aspect's feature with the targeted aspect's feature and then change the output layer to accommodate the new feature dimension of the simplified aggregation window.

Table 9 shows the experimental results. From the performance comparison of simplified aggregation, we observe that the full `LSA` is optimal in most situations, despite the underlying PLM or training dataset. Moreover, to our surprise, `LSA` with "RA" outperforms `LSA` with "LA" in some situations.

Table 9: The average performance deviation of ablated LSA baselines. "LA" and "RA" indicates the simplified aggregating window constructed only exploits the left-adjacent aspect or right-adjacent aspect, respectively.

Model	Laptop14		Restaurant14	
	Acc	F1	Acc	F1
LSA_P-DeBERTa	**84.33(0.37)**	**81.46(0.52)**	**89.91(0.33)**	**84.90(0.49)**
– w/ LA	83.65(0.47)	80.48(0.62)	89.20(0.28)	84.26(0.31)
– w/ RA	83.86(1.25)	80.41(1.26)	88.57(0.65)	83.16(0.78)
LSA_T-DeBERTa	84.16(0.31)	81.40(0.55)	**89.91(0.43)**	**84.96(0.40)**
– w/ LA	84.08(1.25)	81.21(1.51)	89.55(0.62)	84.68(1.13)
– w/ RA	**84.39(0.78)**	**81.54(1.22)**	89.38(0.45)	83.99(0.68)
LSA_S-DeBERTa	**84.33(0.31)**	**81.68(0.44)**	**90.27(0.76)**	**85.78(0.56)**
– w/ LA	83.57(1.10)	80.44(1.14)	89.29(0.89)	84.00(1.22)
– w/ RA	83.95(0.47)	80.89(0.88)	89.55(0.40)	84.26(0.39)

C.4 Experiments on Twitter Dataset

The experimental results on the Twitter dataset reveal that the extended LSA-X models, with LSA_T-X-DeBERTa demonstrating the best performance, effectively leverage local sentiment coherency to achieve competitive accuracy and F1 scores while maintaining consistent results across different runs.

Table 10: The performance of LSA models on the Twitter datasets, and the best results are heightened in **bold**. Numbers in parentheses denote IQR.

Model		Twitter	
		Acc	F1
LSA_P-DeBERTa	LSA	76.91(0.36)	75.90(0.41)
LSA_T-DeBERTa	LSA	76.61(0.20)	76.12(0.27)
LSA_S-DeBERTa	LSA	76.61(0.52)	75.84(0.64)
LSA_P-X-DeBERTa	LSA-X	76.81(0.76)	76.09(0.50)
LSA_T-X-DeBERTa	LSA-X	**77.17(0.71)**	**76.45(0.65)**
LSA_S-X-DeBERTa	LSA-X	77.06(0.26)	76.23(0.29)