Is syntax structure modeling worth? Leveraging pattern-driven modeling to enable affordable sentiment dependency learning

Anonymous ACL submission

Abstract

Is structure information modeling really worth in Aspect-based sentiment classification (ABSC)? Recent popular works tend to exploit syntactic information guiding sentiment dependency parsing, i.e., structure-based sentiment dependency learning. However, many works fall into the trap that confusing the concepts between syntax dependency and sentiment dependency. Besides, structure information (e.g., syntactic dependency tree) usually consumes expensive computational resources due to the extraction of the adjacent matrix. Instead, we believe the sentiment dependency mostly occurs between adjacent aspects. By proposing the sentiment patterns (SP) to boost the sentiment dependency learning, we introduce the **L**ocal dep**en**dency **a**ggregating (Lena) to explore sentiment dependency in the text. Experiments show that Lena is more efficient than existing structure-based models without dependency matrix constructing and modeling expense. The performance on all five public ABSC datasets makes a big step compared to state-of-the-art models, and our work could inspire future research focusing on efficient local sentient dependency modeling.

1 Introduction

In order to solve the absence of explicit sentiment information in the context, recent studies on ABSC (Pontiki et al., 2014) turned to focus on the parsing of sentiment dependency among aspects. For example, `The laptop's storage is large, so does the battery capacity.`, the customer praised both storage and battery capacity, while no direct sentiment description of battery capacity is available in the review. The methods capable of dependency learning can be approximately categorized into the topological structure-based dependency parsing methods (Zhang et al., 2019a; Huang and Carley, 2019), and syntax tree distance-dependent methods(Phan and Ogunbona, 2020). Meanwhile, some works adopt hybrid dependency modeling strategies to enhance the model's ability to learn sentiment dependency. But there are some problem remained in structure modeling. On the one hand, some previous works blurred the gap between syntactical dependency and sentiment dependency, and avoid exploring the relatedness of them. On the other hand, due to the expensive dependency tree parsing time and resources occupation, they are not the ideal solutions for dependency learning in long texts, especially texts with multi-aspects. Table. 1 shows the brief comparison between the structure information-based models and non-structure-based models[1].

Table 1: The resources occupation of state-of-the-art ABSC models. "P.T." and "A.S." indicate the dataset pre-processing time and additional storage requirement, respectively. * represents non-dependency based models, and "†" indicates our models.

Models	Laptop14		Restaurant14	
	P.T. (sec)	A.S. (MB)	P.T. (sec)	A.S. (MB)
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
Lena *†	3.16	0	4.32	0
Lena$_S$ *†	20.56	0	30.23	0

Our study shows that sentiment dependency mostly exists between adjacent aspects, we call this phenomenon "sentiment cluster". We explain the existence of sentiment cluster by introducing sentiment patterns (see Sec. 3.2). This sentiment cluster hypothesis implies the possibility of efficient modeling of sentiment dependency. We exploit this finding by introducing sentiment patterns (SP) to improve ABSC. Meanwhile, we propose a

[1]The experiments are based on RTX 2080 GPU, AMD R5-3600 CPU with PyTorch 1.9.0. The original size of the Laptop14 and Restaurant14 datasets are $336kb$ and $492kb$, respectively.

sentiment dependency learning framework based on sentiment cluster, i.e., the local sentiment dependency aggregating (Lena). Lena handles the sentiment dependency within a local sentiment dependency aggregating window (AW), avoiding the direct modeling of structure information such as trees or graphs. The AW aggregates the aspect-oriented features. Hence, the Lena could be implemented in flexible ways. e.g., we can construct the aspect-oriented features based on attention mechanism, BERT-SPC(Song et al., 2019) or LCF(Zeng et al., 2019) mechanism. We adopt the local context focus (LCF) mechanism to obtain aspect-oriented features and construct aggregating windows. More specifically, we employ the original LCF mechanism and adapted the LCFS(Phan and Ogunbona, 2020) mechanism to implement Lena. Our experimental results show Lena achieve an impressive improvement compared to state-of-the-art models, i.e., up to 86.21% and 91.07% accuracy on the Laptop14 and the Restaurant14 Datasets.

Moreover, Lena is a backbone-free framework, we develop Lena based on several pre-trained models, e.g., BERT(Devlin et al., 2019), RoBERTa(Liu et al., 2019), DeBERTa(He et al., 2021) to evaluate its transferability. Besides, we propose differential weighting for window components to explore better AW construction strategy. The experimental results show that Lena is an efficient sentiment dependency learning method according to the performance comparisons.

Therefore, the main contributions[2] of this paper are as follows:

1 Our research proves that current structure-based modeling methods are not the cure of ABSC. In the contrast, it is highly inefficient and may mislead future research.

2 The novel sentiment patterns are introduced in this paper. We further propose the Lena mechanism for efficient and effective sentiment dependency learning. The experimental results show that our models comprehensively outperform state-of-the-art models without loss of simplicity and efficiency.

3 The differential weighting strategy and simplification strategy is proposed to deeply explore the optimal AW construction strategy of Lena. And we study the effectiveness of and conduct experiments to evaluate the performance of Lena based on multiple pre-trained models.

[2]The code and datasets are available in supplementary material.

2 Related Works

We observe that recent works tend to resolve sentiment dependency problem by modeling syntax-based structures Those structure tree-based methods generally employ the graph convolution network (GCN) and attention mechanism(Bahdanau et al., 2014) to model the sentiment dependency. There are diversities of attention mechanisms proposed in the previous research(Wang et al., 2016; Ma et al., 2017), e.g., multi-grained attention(Zhang et al., 2019a) and multi-head attention(Vaswani et al., 2017). These works ignore the efficiency drawback of syntax tree handling.

Existing popular ABSC methods can be divided into methods based methods based on dependency learning, and methods based on pre-trained models. Meanwhile, some works use the hybrid strategy to improve their works.

2.1 Dependency-based Methods

The researchers have been attempting to model sentiment dependency based on the syntax tree-based structure information which achieves hopeful improvement without consideration of efficiency. Fig. 1 shows an example of a syntax tree that can be used for structure extraction. Early works focus on learning the sentiment dependency based on syntax tree parsed from aspect and context. e.g., (Zhang et al., 2019a) and (Sun et al., 2019) introduce the models based on dependency trees and obtain promising performance. Most ABSC models (Huang and Carley, 2019; Tang et al., 2020; Wang et al., 2020) employ the GCN equipped with an attention mechanism to learn syntactical trees because the GCN can model topological relation and obtains promising performance. Meanwhile, some methods (He et al., 2018; Zhang et al., 2019b) exploit the dependency tree to measure the distance between aspect and context words, those methods avoid modeling the dependency tree directly and have better efficiency. Dai et al. (2021) propose the pre-trained model to induce the dependency tree which can be adapted to several models and achieved state-of-the-art performance. However, this method requires additional and expensive resources (e.g., time and system memory) to induce structure information.

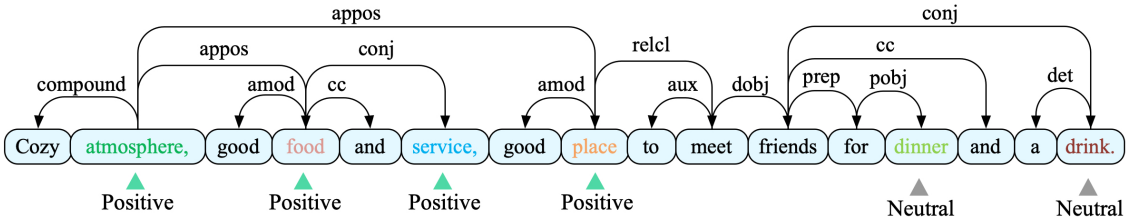

Figure 1: The syntax-tree parsed from a real restaurant review. The colored words are tokens from aspect terms, and the arrowed lines indicate the dependency relations. The dependency matrix built from syntax-tree requires $n \times n$ (n is the length of the text) space to store.

2.2 PTM-based Methods

The pre-trained models prompt the development of ABSC. BERT is one of the first pre-trained models to be applied in ABSC, which achieves exciting performance by fine-tuning without any model modification (Xu et al., 2019). Rietzler et al. (2019) argue that besides fine-tuning, domain adaption of BERT on target corpus could make a great improvement for ABSC. Zhao et al. (2020) and Wang et al. (2020) propose the BERT-based models and exploit dependency trees to learn sentiment information. Scholars recognize that the sentiment polarity of the target aspect is highly related to its local context. Instead of directly modeling dependency tree, Phan and Ogunbona (2020) propose a method that calculates the distance using syntactical information to guide the model to learn the LCF feature and obtain considerable results. There are many other pretrained model-based methods aimed for ABSC in recent years (Tang et al., 2020; Zhou et al., 2020; Li et al., 2021; Silva and Marcacini, 2021), most of them shows hopeful improvement. We do not intend to discuss them in detail but we compare our models with their methods without any evasion.

3 Methodology

Fig. 3 shows the main architecture of the Lena framework. The Lena-based model uses pretrained models to learn the LCF mechanism-based aspect-oriented features of all provided aspects, and the aggregation window travels upon the aspect-oriented features of adjacent aspects. We concatenate the global context feature and aggregation window feature to predict aspect polarity in case of avoiding potential loss of sentiment information outside the aggregation window.

3.1 Preliminaries

Fig.1 shows an example of aspect-based sentiment classification, where "atmosphere", "food" and "service" contain positive sentiment, while "dinner" and "drink" contain neutral sentiment. There may be multiple aspects with different sentiment polarities in a text, and the polarity between each aspect may be dependent or even contradictory.

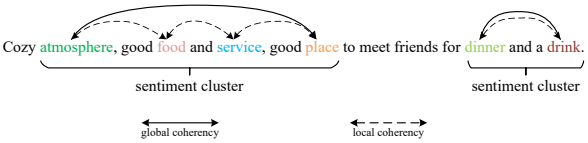

Figure 2: Visualization of the sentiment cluster and sentiment coherency.

3.2 Sentiment Pattern

Inspired by existing works(Zhang et al., 2019a; Zhao et al., 2020) which proved sentiment polarity between aspects is not always independent, we introduce sentiment pattern (SP). i.e., the underlying empirical principles of organization of sentiment polarities, to help the model learn sentiment dependency. Precisely modeling for sentiment patterns may be difficult, we can develop our model under the guidance of SP. We propose two sentiment patterns in this paper and prove our arguments by experiment analysis.

3.2.1 Sentiment Cluster

The aspects containing similar sentiment polarity tend to cluster as shown in Fig 2. As users generally organize the opinions of aspects before giving the review, it is intuitive to realize that users tend to cluster the aspects according to the polarity category. i.e., **SP1**. Table 2 shows the number of aspects belongs to a sentiment cluster with size ≥ 1. We can observe that many of the aspects are clustered.

3.2.2 Sentiment Coherency

Sentiment polarities of multiple aspects are possible to subject to the sentiment coherency as shown

Table 2: The number of aspect in sentiment clusters with different sizes.

Dataset	Cluster Size					Sum
	1	2	3	4	≥5	
Laptop14	791	799	468	294	614	2966
Restaurant14	1318	1050	667	479	1214	4728
Restaurant15	617	406	229	163	326	1741
Restaurant16	836	539	314	210	462	2361
MAMS	6463	2583	1328	746	1397	12517

in Fig 2. In the case of natural thinking style, users are probably to bring up an aspect that has the same polarity as pre-aspect for any thinking pause. The pattern of sentiment coherency can be classified into global and local coherency. We propose our model referring to the local sentiment coherency. i.e., **SP2**.

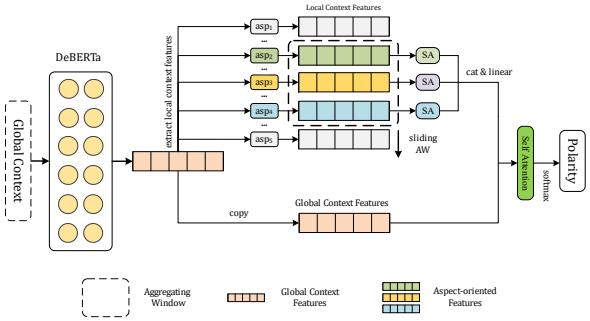

Figure 3: The main framework of Lena.

3.3 Local Sentiment Dependency Aggregating

The Lena is based on **SP1** and **SP2**. The implementation of Lena relies on the aspect-oriented context features. We construct the aggregating window using LCF features of adjacent aspects. i.e., the k-th ($k = 1$ in this paper) left- and right-adjacent aspects are concatenated to be the aggregating window. The calculation of local context can be classified into the relative-position method and syntax distance-based method. In this paper, we employ both methods to extract LCF features and construct the aggregating window[3]. i.e., Lena and Lena$_S$ (Lena-Syntax), respectively.

3.3.1 Relative Distance-based Local Context

Token distance-based local context is calculated using the distance of token-aspect pairs. Assume $W^c = \{w_0^c, w_1^c, \ldots, w_n^c\}$ is the token set after tokenization. The distance $\mathcal{D}_t$ of a token-aspect pairs is calculated as follow:

$$\mathcal{D}_t = \frac{\sum_{i=1}^{m}(p_i - p_t)}{m} \quad (1)$$

[3]see (Phan and Ogunbona, 2020; Yang et al., 2021) for detailed LCF computation

where $p_i (i \in [1, m])$ and p_t are the positions of i-th token within the aspect and the position of any context token, respectively. m is the length of an aspect. It that case, we determine the local context and assign the local context tags according to $\mathcal{D}_t$:

$$T_t = \begin{cases} 0, & \mathcal{D}_t > \alpha \\ 1, & other \end{cases} \quad (2)$$

where n is the length of the tokenized context; $\alpha (\alpha = 3)$ is a fixed threshold to measure local context. Then Lena uses the relative distances to obtain context weights. The Lena applies context weights to the global context feature and obtains the LCF features.

$$H_i^l = \begin{cases} H_i^c & \mathcal{D}_i \leq \alpha \\ 1 - \frac{(\mathcal{D}_i - \alpha)}{n} \cdot H_i^c & \mathcal{D}_i > \alpha \end{cases} \quad (3)$$

Where H_i^c and H_i^l are the hidden states at position i in the global context features and local context features, respectively. This implementation is called Lena.

3.3.2 Syntax Distance-based Local Context

Although directly learning structure tree is inefficient, we can employ the distance calculated from the syntax structure to measure local context and model the local context. Fig. 1 shows a syntax-based tree from a sample with multi-aspects. The distance $\mathcal{D}_t$ can be calculated according to the shortest distance between a token node and aspect nodes in the syntax-based tree. Consistent with the token-based local context calculation method, the syntactic structure-based method also calculates the average distance between the aspect-token and the context token:

$$\mathcal{D}_t = \frac{\sum_{i=1}^{m} min\_dist(t, t_i^{aspect})}{m} \quad (4)$$

where min_dist indicates the shortest distance between i-th token within the aspect and context token t from the non-local context. Similar to the Lena, the Lena$_S$ only replace the token-pair based distance with syntax-node based distance.

3.3.3 Aggregating Window

We use BERT, RoBERTa and DeBERTa as the base models to encode input text. Assume that H^c is the context feature learned from BERT:

$$H_T^l = W_T^l H^c \quad (5)$$

$$H_L^l = W_L H^c \quad (6)$$

$$H_R^l = W_R^l H^c \tag{7}$$

where H_T^l, H_L^l and H_R^l are the LCF features of the target aspect, the feature of left- and right-adjacent aspect. $W_T^l \in \mathbb{R}^{n \times d_h}$, $W_L^l \in \mathbb{R}^{n \times d_h}$ and $W_R^l \in \mathbb{R}^{n \times d_h}$ are the local context weight vectors of aspects. We apply the self-attention for LCF feature of each aspect:

$$H_{SA}^o = [H_{SA}^L, H_{SA}^T, H_{SA}^R] \tag{8}$$

$$H^o = W^o H_{SA}^o + b^o \tag{9}$$

H_{SA}^L, H_{SA}^T, H_{SA}^R are LCF features learned by self-attention. d_h is the dimension of the hidden size and H_{SA}^o is the window composed of concatenated LCF features of multiple adjacent aspects. H^o is the output representation of Lena, W^o and b^o are the trainable weight and bias parameters.

3.3.4 Aggregation Window Padding

We need to pad the aggregation window using the aspect-oriented features. Here are three padding strategy shown in Fig. 4. It is worthy noting that padding sentiment aggregation window does not degenerate model because the padded components are duplicated and the same as edge adjacent aspects. Besides, the padded components have the same sentiment information which is subject to **SP1** and **SP2** while modeling the sentiment clusters.

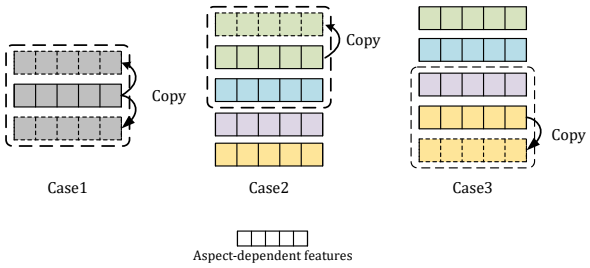

Figure 4: Window padding strategy for different situations.

3.3.5 Differential Weighted Aggregation Window

The Lena treats the sentiment information of adjacent aspects on both left and right sides equally. However, According to **SP2**, it is natural for us to realize that the importance of sentiment information of the left- and right- adjacent aspects are probably different. Thereafter, We propose differential weighting to differential adjust the contribution of sentiment information from the left-adjacent (previous) aspect and the sentiment information of the right-adjacent (following) aspect. Assume η is the adjustable weight of the LCF feature of left and right aspects:

$$H_{att}^{dw} = [\eta H_{SA}^L, H_{SA}^T, (1-\eta) H_{SA}^R] \tag{10}$$

where H_{att}^{dw} is the LCF feature learned through differential weighting Strategy.

3.4 Output Layer

For the purpose of compensating the loss of context feature caused by LCF calculation, we combine the global context feature and feature learned from the local dependency aggregating to predict sentiment polarities as following:

$$O^{fusion} = W^f [H^o, H^c] + b^f \tag{11}$$

$$O^{dense} = W^d O_{head}^{fusion} + b^d \tag{12}$$

$$\hat{y} = \frac{\exp(O^{dense})}{\sum_1^C \exp(O^{dense})} \tag{13}$$

where O_{head}^{fusion} and $\hat{y}$ are the features of first token and predicted sentiment polarity, respectively. C indicates the number of polarity categories. $W^f \in \mathbb{R}^{n \times 2d_h}$, $b^f \in \mathbb{R}^{2d_h}$ and $W^d \in \mathbb{R}^{1 \times C}$, $b^d \in \mathbb{R}^C$ are the trainable weight and bias vectors.

3.5 Model Training

Lena is implemented based on transformers[4], namely Lena-BERT, Lena-RoBERTa, Lena-DeBERTa (a.k.a., Lena) Lena-DeBERTa-Large (a.k.a., LenaX) and we optimize our model using Adam. The objective function is cross-entropy as follows:

$$\mathcal{L} = -\sum_1^C \widehat{y_i} \log y_i + \lambda \|\Theta\|_2 \tag{14}$$

where λ and Θ are the L_2 regularization and parameter set of the model.

4 Experiments

4.1 Datasets and Hyper-parameters

To comprehensively evaluate the performance of the local dependency aggregating mechanism, we conducted experiments on five public

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

datasets[5] (containing multiple aspects): the Laptop14 and Restaurant14 datasets from SemEval-2014 Task4(Pontiki et al., 2014), the Restaurant15, Restaurant16 datasets from SemEval-2015 task12(Pontiki et al., 2015), SemEval-2016 task5(Pontiki et al., 2016) and MAMS datasets from (Jiang et al., 2019), respectively.

Table 3: The statistics of five datasets used in this work.

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

We fine-tune the hyper-parameter settings on the datasets. The learning rate of Lena is 1e-5. The batch size and maximum text length are 16 and 80, respectively. The L_2 regularization parameter λ is 1e-8, and the local context threshold α is 3 for both Lena and Lena$_S$. Each model trained for five rounds and the median performance is presented.

4.2 Overall Performance

We compare the performance of Lena and Lena$_S$ with recent state-of-the-art models without any evasion (many of them are structure-based dependency learning methods). However, we do not intend to introduce them separately, please see the original paper refer to Table 4.

Table 4 shows the main experimental results. Overall, the Lena baseline model obtain substantial improvements over most of the BERT- and RoBERTa-based Lena models on all five datasets. In particular, the LenaX$_S$ achieves the un-parallelable improvement compared to existing methods, no matter structure learning methods or recent DeBERTa models. Compared with the DeBERT-based model, LenaX and Lena significantly outperform approximately 1.5%, 3.5% accuracy on all four datasets. As for comparisons with structure-based learning method GCN-based SDGCN and SK-GCN-BERT, Lena-based models significantly improve the classification accuracy and F1 on all five datasets. What impresses us is that the distance-based Lena arrives comparable metric compared to Lena$_S$, while the latter requires more data pre-processing time, i.e., approximate 8 multiples compare to the former. We do not need to make more performance comparisons, as the metrics are fair and stand for themselves. Combined with Fig. 1, We have a conclusion that Lena models perform impressively in handling local sentiment dependency without any GCN architecture or additional structure matrix.

[5]The processed datasets are available with the code in supplementary materials.

4.3 Differential Weighting for AW

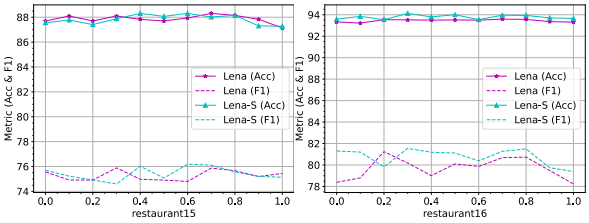

Figure 5: Visualization of average performance exploiting differential weighting on the Restaurant15 and Restaurant16 datasets.

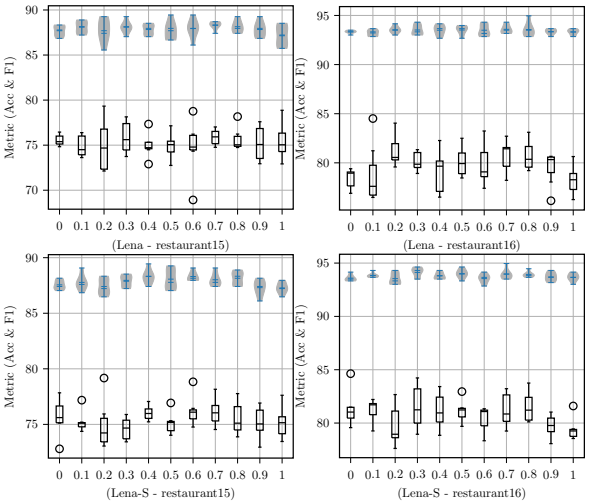

Figure 6: Visualization of performance under differential weighting (η) on the Restaurant15, Restaurant16 dataset. The violin plot and box plot parts are Accuracy metric and F1 metric, respectively.

It is natural to realize that the order of aspects in the text matters while modeling the aggregation window. Because users tend to comment on an aspect that has the same polarity as the pre-commented aspects. We design the differential-weighting to model this effect. We use η ($\eta \in [0, 1]$) to adjust the contribution of LCF features of side aspects. A greater η means more contribution of the left-aspect's LCF feature. Fig 5 and Fig 6 show the performance of the model under different η.

It is clear to observe that the contribution of adjacent aspects on left- and right- sides are differ-

Table 4: The overall performance of Lena variants on five datasets. "Dep" means structure information modeling. The best experimental results are heightened in **bold**. All experiments results of Lena are medians in five runs.

Model	Dep.	Laptop		Restaurant14		Restaurant15		Restaurant16		MAMS	
		Acc	F1	Acc	F1	Acc	F1	Acc	F1	Acc	F1
BERT (Devlin et al., 2019)	N	79.73	75.5	82.74	73.73	82.16	64.97	89.43	74.2		
BERT-PT (Xu et al., 2019)	N	78.89	75.89	85.92	79.12	-	-	-	-	-	-
SDGCN-BERT (Zhao et al., 2020)	Y	81.35	78.34	83.57	76.47	-	-	-	-	-	-
CapsNet-BERT (Jiang et al., 2019)	N	-	-	-	-	-	-	-	-	83.39	80.91
LCF-BERT (Zeng et al., 2019)	N	79.73	76.07	86.16	80.12	83.77	69.03	91	77.1	-	-
RGAT-BERT (Bai et al., 2020)	Y	80.94	78.2	86.68	80.92	-	-	-	-	-	-
SK-GCN-BERT (Zhou et al., 2020)	Y	79	75.57	83.48	75.19	83.2	66.78	87.19	72.02	-	-
DGEDT-BERT (Tang et al., 2020)	Y	79.8	75.6	86.3	80	84	71	91.9	79	-	-
LCFS-BERT (Phan and Ogunbona, 2020)	N	80.25	76.72	86.43	80.84	84.07	69.67	90.35	76.28	-	-
ASGCN-BERT (Zhang et al., 2019a)	Y	79.83	75.89	84.76	77.94	84.22	72.9	91.05	77.05	-	-
BERTAsp+SCAPT (Li et al., 2021)	N	82.76	79.15	89.11	83.79	-	-	-	-	-	-
RGAT-RoBERTa (Dai et al., 2021)	Y	83.33	79.95	87.52	81.29	-	-	-	-	-	-
PWCN-RoBERTa (Dai et al., 2021)	Y	84.01	81.08	87.35	80.85	-	-	-	-	-	-
ABSA-DeBERTa (Silva and Marcacini, 2021)	N	82.76	79.36	89.46	83.42	-	-	-	-	-	-
Lena (Ours)	N	84.16	81.4	90.03	85.92	88.15	77.07	93.82	79.93	84.96	84.41
Lena$_S$ (Ours)	N	84.33	80.97	89.64	84.08	89.04	78.54	94.47	84.84	85.18	84.62
LenaX (Ours)	N	86.13	83.36	90.31	85.5	90	78.46	**95.2**	**84.8**	**85.7**	**85.21**
LenaX$_S$ (Ours)	N	**86.21**	**83.87**	**91.07**	**86.38**	**90.74**	**78.79**	94.8	84.31	85.55	85

ent. However, because the datasets are small and contain error data, our experiment shows different optimal η for Lena variants considering Accuracy metric. Hopefully, while $\eta \in [0.7, 0.8]$, we observe a general good performance in most situations. On the other hand, the fixed hyperparameter η is hard to precisely measure the significance of the sentiment information of side aspects. We will consider adaptive η calculation methods in the future.

The difference between simplified AW (SAW) and differential-weighting AW (DAW) with $\eta = 0$ or $\eta = 1$ is the network structure, as the DAW employs a full-connected layer with $3 \times d_h$ input size ($2 \times d_h$ in SAW learning the window features.

4.4 Decomposition of Lena

Table 5 shows the ablation experimental results. From the performance of simplified Lena, and BERT-, RoBERTa-based Lena variants, we see a certain clue that the DeBERTa baseline is better in most situations. Compared with the full Lena, although the simplified Lena slightly improves the sentiment classification efficiency. In most situations the simplified Lena suffers a loss of performance inevitably. Moreover, the Lena with LA usually performs better than Lena with RA, which is similar to the conclusion in DAW analysis: the better η usually lies between $(0.5, 1)$ (a typical ideal η is 0.7).

4.5 Research Questions

RQ1: Does the structure and GCN based method learn the sentiment dependency?

According to Table. 4 and our analysis, the answer may be yes but not necessary. The GCN-based methods rely on the syntax tree or other topological information. However, the are some limitations that remain unresolved.

On the one hand, measuring the importance of sentiment dependency between different aspects is very difficult. Most existing works confuse the border of syntax dependency and sentiment dependency. i.e., assuming the syntax dependency may help sentiment dependency learning implicitly. But this concept approximate does not outperform the aspect-focused modeling method. e.g., LCF-BERT. We only focus on the local sentiment dependency learning based on sentiment patterns, which prevents the inefficient coarse-grained structure matrix modeling. And experiment results show that the Lena outperforms state-of-the-art models without loss of simplicity and effectiveness. So the GCN may make merely sense to some extent.

On the other hand, the syntax trees are parsed based on the existing tool which does not subject to the same tokenization strategy. In that case, there are many inevitable alignment problems between syntax tree nodes and deep learning tokenizers. Although Dai et al. (2021) propose to use the pre-trained model to induce the dependency tree which alleviates alignment problem, this method requires non-negligible expense. Thereafter, we believe the

Table 5: The average performance deviation of ablated Lena baselines. "LA" and "RA" indicates the simplified aggregating window constructed only exploits the left-adjacent aspect or right-adjacent aspect, respectively.

Model	Dep.	Laptop		Restaurant14		Restaurant15		Restaurant16		MAMS	
		Acc	F1	Acc	F1	Acc	F1	Acc	F1	Acc	F1
Lena	N	84.16	81.4	90.03	85.92	88.15	77.07	93.82	79.93	84.96	84.41
w/ LA	N	81.92	80.2	89.2	84.62	88.51	76.96	92.36	78.83	84.81	84.49
w/ RA	N	83.54	80.03	89.55	85.13	**89.44**	77.24	93.01	79.99	85.03	84.61
w/ BERT	N	80.88	77.27	86.25	80.14	84.69	71.97	91	77.44	84.73	84.21
w/ RoBERTa	N	84.17	**81.69**	88.84	83.79	87.22	75.86	93.82	83.15	84.51	83.93
Lena$_S$	N	**84.33**	80.97	89.64	84.08	89.04	**78.54**	**94.47**	**84.84**	**85.18**	**84.62**
w/ LA	N	82.92	79.96	**90.54**	**86.17**	88.52	77.27	93.98	81.57	84.73	84.34
w/ RA	N	83.7	80.36	89.11	83.79	87.41	77.21	93.5	79.85	**85.18**	82.79
w/ BERT	N	81.35	78.35	87.14	81.04	84.81	72.21	92.2	79.5	85.05	84.31
w/ RoBERTa	N	83.55	80.89	88.12	82.76	87.96	75.42	93.66	80.47	84.73	84.15

local sentiment dependency learning can prompt the ABSC research.

RQ2: Are there other ways to build an AW?

Lena is a paradigm rather than a hard network structure, which means Lena is extensible and flexible. The Aggregating Window is the core of Lena, which is composed of aspect-oriented feature vectors. We adopt the LCF mechanism to build the AW, and this initial implementation yields remarkable performance improvements. However, it is worth noting that there are many alternative strategies to build the AW. We also tried to construct $[CLS] + Context + [SEP] + Aspect + [SEP]$ to learn the aspect-oriented feature vectors, and concatenate the vectors as AW, and also achieved promising performance. That is, aspect-oriented features derived from any method are available for constructing AW. e.g., some well-designed attention mechanisms.

RQ3: What are the differences between convolution and AW?

There may be confusion that aggregating window is a special case of convolutional structure. However, convolution and AW are totally different in concept. The main differences are as follows:

Modeling target. Convolution is a continuous in-modeling strategy that is usually used for token-level feature learning, e.g., learning embedded text representation. However, the AW is a discrete post-processing strategy for output feature vectors from a neural network.

Processing granularity. The essence of AW is a concatenated feature vector, each component of AW is a vector with the same size of context-level feature vectors. While convolution network is used to extract context-level features. That means the convolution network can be used as the backbone model of Lena. i.e., what Lena aims for are what convolution network can't do.

4.6 The threatening of local sentiment aggregating

Although Lena achieves impressive performance for multiple-aspects situations, e.g., SemEval-2014 datasets. However, while being applied in mono aspect situations, Lena would be degenerated to be a local context focus variant model.

On the other hand, the parsing quality of syntax trees affects the extraction of LCF features. We use spaCy to obtain the syntax tree for Lena$_S$ as in previous works. Due to the alignment problem of tokenization between spaCy and pre-trained models (word-piece and sentence-piece). i.e., there are considerable samples among five datasets that are tokenized into different token set in spaCy and sentence-piece, respectively. In that case, there is a non-negligible error rate in calculating aspect-token pair distance and extracting LCF features. Considering the Lena is $7 \sim 10$ times faster in the data pre-processing procedure, it would be a prior choice in most situations.

5 Conclusion

We argue that structure-based sentiment learning is inefficient and not necessary. By introducing sentiment patterns, we propose the Lena models which use the aggregating window to learn the local sentiment dependency. Compared with the dependency tree-based models, the Lena models only exploit a few distance information and achieve impressive performance on all five datasets. Compared to the state-of-the-art models, Lena also outperforms in five commonly used datasets without loss of efficiency and simplicity. Moreover, we also propose differential weighting for AW to measure the impor-

tance of sentiment information of different aspects. Our work indicates that focusing on the local sentiment dependency learning is an important method to prompt ABSC research. In the future, we plan to work on other window construction methods and propose a self-adaptive differential weighting method to improve the performance of Lena.

References

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*.

Xuefeng Bai, Pengbo Liu, and Yue Zhang. 2020. Investigating typed syntactic dependencies for targeted sentiment classification using graph attention neural network. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 29:503–514.

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. *arXiv preprint arXiv:2104.04986*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186.

Pengcheng He, Jianfeng Gao, and Weizhu Chen. 2021. Debertav3: Improving deberta using electra-style pretraining with gradient-disentangled embedding sharing. *arXiv preprint arXiv:2111.09543*.

Ruidan He, Wee Sun Lee, Hwee Tou Ng, and Daniel Dahlmeier. 2018. Effective attention modeling for aspect-level sentiment classification. In *Proceedings of the 27th international conference on computational linguistics*, pages 1121–1131.

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)*, pages 5472–5480.

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)*, pages 6280–6285.

Zhengyan Li, Yicheng Zou, Chong Zhang, Qi Zhang, and Zhongyu Wei. 2021. Learning implicit sentiment in aspect-based sentiment analysis with supervised contrastive pre-training. *arXiv preprint arXiv:2111.02194*.

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

Dehong Ma, Sujian Li, Xiaodong Zhang, and Houfeng Wang. 2017. Interactive attention networks for aspect-level sentiment classification. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence*, pages 4068–4074.

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*, pages 3211–3220.

Maria Pontiki, Dimitrios Galanis, Haris Papageorgiou, Ion Androutsopoulos, Suresh Manandhar, Mohammad Al-Smadi, Mahmoud Al-Ayyoub, Yanyan Zhao, Bing Qin, Orphée De Clercq, et al. 2016. Semeval-2016 task 5: Aspect based sentiment analysis. In *International workshop on semantic evaluation*, pages 19–30.

Maria Pontiki, Dimitrios Galanis, Harris 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 2015)*, pages 486–495.

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 2014)*, pages 27–35, Dublin, Ireland. Association for Computational Linguistics.

Alexander Rietzler, Sebastian Stabinger, Paul Opitz, and Stefan Engl. 2019. Adapt or get left behind: Domain adaptation through bert language model finetuning for aspect-target sentiment classification. *arXiv preprint arXiv:1908.11860*.

Emanuel H Silva and Ricardo M Marcacini. 2021. Aspect-based sentiment analysis using bert with disentangled attention. In *Proceedings of the LatinX in AI (LXAI) Research workshop at ICML 2021*.

Youwei Song, Jiahai Wang, Tao Jiang, Zhiyue Liu, and Yanghui Rao. 2019. Attentional encoder network for targeted sentiment classification. *arXiv preprint arXiv:1902.09314*.

Kai Sun, Richong Zhang, Samuel Mensah, Yongyi Mao, and Xudong Liu. 2019. Aspect-level sentiment analysis via convolution over dependency tree. 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), pages 5683–5692.

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*, pages 6578–6588.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, pages 6000–6010.

Kai Wang, Weizhou Shen, Yunyi Yang, Xiaojun Quan, and Rui Wang. 2020. Relational graph attention network for aspect-based sentiment analysis. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 3229–3238.

Yequan Wang, Minlie Huang, Xiaoyan Zhu, and Li Zhao. 2016. Attention-based lstm for aspect-level sentiment classification. In *Proceedings of the 2016 conference on empirical methods in natural language processing*, pages 606–615.

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

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.

Biqing Zeng, Heng Yang, Ruyang Xu, Wu Zhou, and Xuli Han. 2019. Lcf: A local context focus mechanism for aspect-based sentiment classification. *Applied Sciences*, 9(16):3389.

Chen Zhang, Qiuchi Li, and Dawei Song. 2019a. 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)*, pages 4560–4570.

Chen Zhang, Qiuchi Li, and Dawei Song. 2019b. Syntax-aware aspect-level sentiment classification with proximity-weighted convolution network. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1145–1148.

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

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. *Knowledge-Based Systems*, 205:106292.

This is an appendix.