Discrete Opinion Tree Induction for Aspect-based Sentiment Analysis

Chenhua Chen[1,2], **Zhiyang Teng**[1,2], **Zhongqing Wang**[3] and **Yue Zhang**[1,2]
[1]School of Engineering, Westlake University, China
[2]Institute of Advanced Technology, Westlake Institute for Advanced Study,
[3]Soochow University,
{chenchenhua, tengzhiyang}@westlake.edu.cn,
wangzq@suda.edu.cn, yue.zhang@wias.org.cn

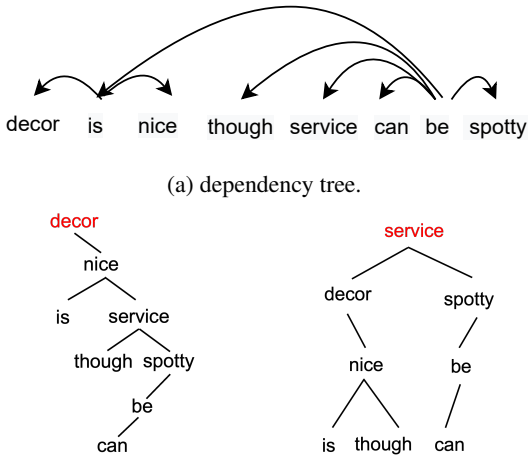
Abstract

Dependency trees have been intensively used with graph neural networks for aspect-based sentiment classification. Though being effective, such methods rely on external dependency parsers, which can be unavailable for low-resource languages or perform worse in low-resource domains. In addition, dependency trees are also not optimized for aspect-based sentiment classification. In this paper, we propose an aspect-specific and language-agnostic discrete latent opinion tree model as an alternative structure to explicit dependency trees. To ease the learning of complicated structured latent variables, we build a connection between aspect-to-context attention scores and syntactic distances, inducing trees from the attention scores. Results on six English benchmarks, one Chinese dataset and one Korean dataset show that our model can achieve competitive performance and interpretability.

1 Introduction

Aspect-based sentiment classification (ABSA) is the task of recognizing the sentiment polarities of specific aspect categories or aspect terms in a given sentence (Jiang et al., 2011; Dong et al., 2014; Wang et al., 2016; Tang et al., 2016; Li et al., 2018; Du et al., 2019; Sun et al., 2019a; Seoh et al., 2021; Xiao et al., 2021). Different from document-level sentiment analysis, different aspect terms in the same document can bear different sentiment polarities. For example, given a restaurant review "**decor** is nice though **service** can be spotty", the corresponding sentiment labels of "decor" and "service" are positive and negative, respectively.

How to locate the corresponding opinion contexts for each aspect term is a key challenge for ABSA. To this end, recent efforts leverage dependency trees (Zhang et al., 2019; Sun et al., 2019a; Wang et al., 2020). Syntactic dependencies have been shown to better capture the interaction between the aspect and the opinion contexts (Huang et al., 2020; Tang et al., 2020). For example, in Figure1(a), using syntactic relations, we can find that the corresponding opinion words for "decor" and "service" are "nice" and "spotty", respectively.



(a) dependency tree.

(b) Induced tree for "decor". (c) Induced tree for "service".

Figure 1: A dependency tree of the input sentence "**decor** is nice though **service** can be spotty" and two induced trees of two aspects in this sentence.

Despite its effectiveness, dependency syntax has the following limitations. First, dependency parsers can be unavailable for low-resource languages or perform worse in low-resource domains (Duong et al., 2015; Rotman and Reichart, 2019; Vania et al., 2019; Kurniawan et al., 2021). Second, dependency trees are also not *optimized* for aspect-based sentiment classification. Previous studies transform dependency trees to aspect-specific forms by hand-crafted rules (Dong et al., 2014; Nguyen and Shirai, 2015; Wang et al., 2020) to improve the aspect sentiment classification performance. However, the tree structure is adjusted mainly by the node hierarchy, without optimizing dependency relations for ABSA.

In this paper, we explore a simple method to induce a discrete opinion tree structure automatically for each aspect. Two examples are shown in Figure 1. In particular, given a target and a sentence,

Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics
Volume 1: Long Papers, pages 2051 - 2064

our algorithm induces a tree structure recursively according to a set of attention scores, calculated using a neural layer on top of BERT representation of the sentence (Devlin et al., 2019). Starting with the root node, the algorithm builds a tree by selecting one child node on each side of a current node and recursively continue the partition process to obtain a binarized and lexicalized tree structure. The resulting tree serves as the input structure and is fed into graph convolutional networks (Kipf and Welling, 2017) for learning the sentiment classifier. We study policy-based reinforcement learning (Williams, 1992) to train the tree inducer. One challenge is that the generated policy can be easily remembered by the BERT encoder, which leads to insufficient explorations (Shi et al., 2019). To alleviate this issue, we propose a set of regularizers to help BERT-based policy generations.

Although our method is conceptually simple and straightforward for the inference stage, we show that it has a deep theoretic grounding. In particular, the attention based tree induction parsers trained using the policy network can be viewed as a simplified version to a standard latent tree structured VAE model (Kingma and Welling, 2014; Yin et al., 2018), where the KL divergence between the prior and the posterior tree probabilities is approximated by attention-based syntactic distance measures (Shen et al., 2018a).

Experiments on six English benchmarks, a Chinese hotel review dataset and a Korean automotive review dataset show the effectiveness of our proposed models. The discrete structure also makes it easy to interpret the classification results. In addition, our algorithm is faster, smaller and more accurate than a full variational latent tree variable model. To our knowledge, we are the first to learn aspect-specific discrete opinion tree structures with BERT. We make our code publicly available at https://github.com/CCSoleil/dotGCN.

2 Model

Figure 2 shows the architecture of our proposed model. Given an input sentence x and a specific aspect term a, we induce an opinion tree t according to a recognition network $Q_\phi(t|x,a)$, where ϕ is the set of network parameters. We apply multi-layered graph convolutional networks (GCNs) over the BERT output vectors to model the structural relations in the opinion tree and extract aspect-specific features. Finally, we use an attention-based classifier to learn the sentiment classifier $P_\theta(y|x,a,t)$, where θ is the set of parameters.

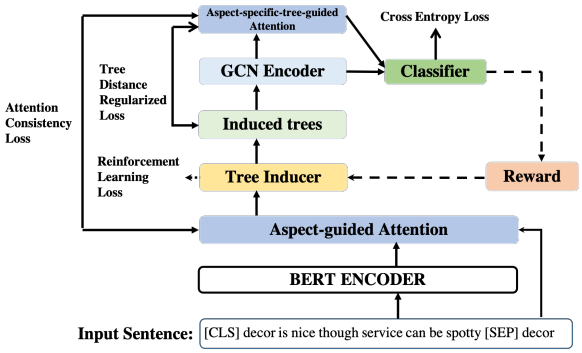


Figure 2: The model architecture.

To train the model, RL is used for $Q_\phi(t|x,a)$ (Section 2.3) and standard backpropagation is used for training $P_\theta(y|x,a,t)$ (Section 2.2).

2.1 Opinion Tree Based Classifier

Opinion Tree Denote the input sentence as $x = w_1 w_2 \dots w_n$ and the aspect as $a = w_b w_{b+1} \dots w_e$. $[b,e]$ is a continuous span of $[1,n]$. w_i is the i-th word. As shown in Figure 1, the opinion tree for a is a binarized tree. Each node contains a word span and at most two children. a is placed at the root node. Except for the root node, [1] each node contains only one word. An in-order traversal over t can recover the original sentence. Ideally, the nodes near the root node should contain the corresponding opinion words, such as "nice" for "decor" and "spotty" for "service".

Algorithm 1 shows the process of building an opinion tree t for a that conforms to the above conditions using a node score function $\mathbf{v}$, where $\mathbf{v}_i$ indicates the informative score of the i-th word contributing to the sentiment polarity y of a. $\mathbf{v}_i^j$ is the corresponding scores of words in the span $[i,j]$. We first make the aspect span $[b,e]$ as the root node and then build its left and right children from the spans $[1,b-1]$ and $[e+1,n]$, respectively. To build the left or right subtree, we first select the element with the largest score in the span as the root node of the subtrees and then recursively use the *build_tree* call for the corresponding span partitions.

Calculating v Following Song et al. (2019), we feed the inputs "[CLS] w_1 w_2 $\dots w_n$ [SEP] w_b $w_{b+1} \dots w_e$" to BERT[2] to obtain the aspect-specific sentence representation $\mathbf{H}$, and then calculate a set

[1] A case study in Appendix shows an example of a root node containing multiple words "grilled alaskan king salmon".

[2] To obtain word-level representations by BERT, we average the output vectors of the corresponding subword tokens.

Input: The scores $\mathbf{v}_1^n$, the aspect span $[b, e]$;
//build the root node ;
root ← new TreeNode;
root.words = $w_b w_{b+1} \dots w_e$; //w_i is the i-th word.
root.left = build_tree ($\mathbf{v}_1^{b-1}$, 1, b-1);
root.right = build_tree($\mathbf{v}_{e+1}^n$, e+1, n);
build_tree($\mathbf{v}_i^j$, i, j):
  if $i > j$: return None;
  node ← new TreeNode;
  $k \leftarrow \arg\max_{k' \in [i,j]} \mathbf{v}_{k'}$;
  node.words = w_k;
  node.left = build_tree($\mathbf{v}_i^{k-1}$, i, $k-1$);
  node.right = build_tree($\mathbf{v}_{k+1}^j$, $k+1$, j);
  return node;
Output: $root$;

Algorithm 1: Aspect-specific construction algorithm given a scoring function $\mathbf{v}$.

of attention scores of the aspect words,

$$\mathbf{v}^p = \mathbf{u}_p \sigma(\mathbf{W}_p \mathbf{H} + \mathbf{W}_{a,p} \mathbf{h}_a),\ \mathbf{s}^p = \text{softmax}(\mathbf{v}^p), \quad (1)$$

where $\mathbf{u}_p$, $\mathbf{W}_p$ and $\mathbf{W}_{a,p}$ are model parameters, σ is the ReLU activation function, $\mathbf{h}_a$ is the aspect representation by sum pooling from $\mathbf{H}_b \mathbf{H}_{b+1} \dots \mathbf{H}_e$. ϕ in $Q_\phi(t|x,a)$ contains the model parameters of BERT, $\mathbf{u}_p$, $\mathbf{W}_p$ and $\mathbf{W}_{a,p}$.

Graph Representation Given t and $\mathbf{H}$, we use GCNs to learn the representation vectors for each word. We convert t to an undirected graph G. Specifically, we take each word as a node in G and design the adjacency matrix $\mathbf{A} \in \mathbb{R}^{n \times n}$ of G by considering four types of edges. First, we include self loops for each word. Second, we fully connect each word within the aspect term. Third, for the child node w_j of the root node, we link w_j to each word in a. Last, we consider edges in t between single word nodes except the root node. Formally, $\mathbf{A}$ is given by

$$\mathbf{A}_{i,j} = \begin{cases} 1 & \text{if } i = j, \ \text{(self-loops)} \\ 1 & \text{if } i \in (b,e) \text{ and } j \in (b,e), \ \text{(aspect words)} \\ 1 & \text{if } i \in [b,e] \text{ and } a \text{ is the parent node of } w_j \\ 1 & \text{if } w_i \text{ is the parent or a child node of } w_j \\ 0 & \text{otherwise.} \end{cases} \quad (2)$$

$\mathbf{A}$ is ensured to be symmetric by Eq 2.

We then use GCNs to capture the structured relations between word pairs, given the adjacency matrix $\mathbf{A}$ between nodes and the representation matrix of the $(l-1)$-th layer $\mathbf{H}^{l-1} \in \mathbb{R}^{n \times d}$, the l-th layer representation $\mathbf{H}^l$ given by a GCN is,

$$\mathbf{H}^l = f(\mathbf{A}\mathbf{H}^{l-1}\mathbf{W}^l + \mathbf{b}^l), \quad (3)$$

where f is an activation function (i.e., ReLU), $\mathbf{W}^l \in \mathbb{R}^{d \times d}$ and $\mathbf{b}^l \in \mathbb{R}^d$ are the model parameters for the l-th layer. The input to the first GCN layer $\mathbf{H}^0$ is $\mathbf{H}$ given by the sentence encoder.

Target Aspect Representation We consider both the representation vector of the "[CLS]" token ($\mathbf{H}_{cls}^0$) and the aspect vectors given by the last GCN layer ($\mathbf{H}_b^N, \mathbf{H}_{b+1}^N \dots, \mathbf{H}_e^N$) as the aspect-specific representation vector to query the input sentence representation $\mathbf{H}^0$. The final aspect-specific feature representation $\mathbf{c}$ over the input sentence representation is given by an attention layer,

$$\alpha_t = (\mathbf{H}_t^0)^T (\mathbf{H}_{cls}^0 + \sum_{i=b}^{e} \mathbf{H}_i^N),\ \boldsymbol{\alpha} = \text{softmax}(\boldsymbol{\alpha}),\ \mathbf{c} = \boldsymbol{\alpha}\mathbf{H}^0, \quad (4)$$

where α_t is the attention scores of a to w_t, $\boldsymbol{\alpha}$ is the normalized scores and $\mathbf{c}$ is the final feature.

Output layers use $\mathbf{c}$ for computing the sentiment polarity scores. The final sentiment distribution is given by a softmax classifier,

$$\mathbf{p} = \text{softmax}(\mathbf{W}_c \mathbf{c} + \mathbf{b}_c), \quad (5)$$

where $\mathbf{W}_c$ and $\mathbf{b}_c$ are model parameters and $\mathbf{p}$ is the predicted distribution.

2.2 Training the Sentiment Classifier

Cross Entropy Loss The classifier is trained by maximizing the log-likelihood of the training samples. Formally, the objective is to minimize

$$\mathcal{L}_{\text{sup}} = -\sum_{i=1}^{|D|} \sum_{a \in x_i} \log \mathbf{p}_{i,y_a}, \quad (6)$$

where $|D|$ is the size of training data, y_a is the sentiment label of a in the i-th example x_i and $\mathbf{p}_{i,y_a}$ is the classification probability for a, which is given by Eq 5. The set of model parameters θ in $P_\theta(y|x,a,t)$ includes GCN blocks and the classifier parameters in Eq 5.

Tree Distance Regularized Loss Following Pouran Ben Veyseh et al. (2020), we introduce a syntax constraint to regularize the attention weights. Ideally, the words near to the root node should receive high attention weights. Given an opinion tree t, we compute the tree distance d_i for each word i using the length of the shortest path to the root. Given the distances and the attention scores $\boldsymbol{\alpha}$, we use the KL divergence to encourage the aspect term to attend the contexts with shorter distances.

$$\begin{aligned} \mathbf{td} &= \text{softmax}([-d_1, \dots, -d_i, \dots, -d_n]) \\ \mathcal{L}_{td} &= \text{KL}(\mathbf{td}, \boldsymbol{\alpha}), \end{aligned} \quad (7)$$

where td_i is the normalized tree distance and KL is the Kullback-Leibler (KL) divergence.

Backpropagation During training, we replace the argmax operator in Algorithm 1 with stochastic

sampling to explore more discrete structures. Since the tree sampling process is a discrete decision making procedure, it is non-differentiable. The gradient can be propagated from $\mathcal{L}_{\text{sup}}$ in Eq 6 to t and θ, but can not be further propagated from t to ϕ. Therefore, we use the policy gradient given by REINFORCE (Williams, 1992) to optimize ϕ of the policy network (Section 2.3).

2.3 Training the Tree Inducer

Suppose that the reward function for a latent tree t is R_t, the goal of reinforcement learning is to minimize the negative expected reward function,

$$\mathcal{L}_{\text{rl}} = -\mathbb{E}_{Q_\phi(t|x,a)} R_t \tag{8}$$

For each t, we use the sentiment log-likelihood $\log P_\theta(y|x,t,a)$ as R_t. Using REINFORCE, the gradient of L_{rl} with respect to ϕ is,

$$\frac{\partial \mathcal{L}_{\text{rl}}}{\partial \phi} = -\mathbb{E}_{Q_\phi(t|x,a)}[R_t \frac{\partial \log Q_\phi(t|x,a)}{\partial \phi}] \tag{9}$$

$\log Q_\phi(t|x,a)$ is the log-likelihood of the generated sample t, which can be decomposed to a sum of log-likelihood at each tree-building step. According to Algorithm 1, each call of *build_tree*$(\mathbf{v}_i^j, i, j)$ involves selecting an action k from the span $[i,j]$ given the scores $\mathbf{v}_m^n$. The action space contains $j-i+1$ actions. The log-likelihood of this action is given by,

$$\log \pi_k = \log \frac{\exp(\mathbf{v}_k)}{\sum_{l=i}^{j} \exp(\mathbf{v}_l)}, \quad i \le k \le j. \tag{10}$$

In particular, we use $\mathbf{v}^p$ in Eq 1 as the score function $\mathbf{v}$. Enumerating all possible trees to calculate the expectation term in Eq 9 is intractable, and we use a Monte Carlo method (Rubinstein and Kroese, 2016), approximating the training objective by taking M samples,

$$\begin{aligned} &\mathbb{E}_{Q_\phi(t|x,a)}[R_t \frac{\partial \log Q_\phi(t|x,a)}{\partial \phi}] \\ &\approx \frac{1}{M} \sum_{i=1}^{M} R_{t_i} \frac{\partial \log Q_\phi(t_i|x,a)}{\partial \phi}. \end{aligned} \tag{11}$$

Attention Consistency Loss Instead of solely relying on the reinforced gradient to train the policy network, we also apply an attention consistency loss to directly supervise the policy network. Note that there are two attention scores in our model. The first is the attention score $\mathbf{s}^p$ defined in Eq 1, which is trained by the reinforcement learning algorithm. The second is the attention score $\boldsymbol{\alpha}$ defined in Eq 4 for extracting useful context features for the aspect-specific classifier. $\boldsymbol{\alpha}$ is trained via end-to-end back propagation. Intuitively, words that receive the largest attention scores should be effective opinion words of the target aspect. Therefore, it should be put closer to the root node by the policy network. To this end, we enforce a consistent regularization between the two attention scores so that polarity oriented attention $\boldsymbol{\alpha}$ can be directly used to supervise the scoring policy $\mathbf{s}^p$. Formally, $\mathcal{L}_{att}$ is given by,

$$\mathcal{L}_{att} = \text{KL}(\boldsymbol{\alpha}.\text{detach}(), \mathbf{s}^p), \tag{12}$$

where *detach* is a stop gradient operator.

Overall Loss Finally, the overall loss is given by

$$\mathcal{L} = \mathcal{L}_{sup} + \lambda_{rl}\mathcal{L}_{rl} + \lambda_{att}\mathcal{L}_{att} + \lambda_{td}\mathcal{L}_{td}, \tag{13}$$

where $\mathcal{L}_{sup}$ is the supervised loss, $\mathcal{L}_{rl}$ is the reinforcement learning loss, $\mathcal{L}_{att}$ is a novel attention consistency loss and $\mathcal{L}_{td}$ is a loss to guide the attention score distributions by tree constraints. λ_{rl}, λ_{att} andλ_{td} are hyper-parameters.

3 A Variational Inference Perspective

Interestingly, $\mathcal{L}_{sup}$, $\mathcal{L}_{rl}$ and $\mathcal{L}_{att}$ can be unified in a theoretic framework using variational inference (Kingma and Welling, 2014). We show in this section, that our method can be viewed as a stronger extension to a latent tree VAE model.

3.1 Variational Latent Tree Model

To model $P_\theta(y|x,a)$, we introduce a latent discrete structured variable t. Formally, the training objective is to minimize the negative log-likelihood,

$$\mathcal{L}_{\text{MLE}} = -\log P(y|x,a,\theta) = -\log \sum_t P_\theta(y,t|x,a), \tag{14}$$

Eq 14 calculates log-of-sum over all possible trees t, which is exponential. Eq 14 can be approximated by the evidence lower bound (ELBO) using variational parameters ϕ (Kingma and Welling, 2014; Yin et al., 2018),

$$\begin{aligned} \mathcal{L}_{\text{ELBO}} = &-\mathbb{E}_{q_\phi(t|x,y,a)}[\log P_\theta(y|x,a,t)] \\ &+ \text{KL}\Big(q_\phi(t|x,y,a), p_\theta(t|x,a)\Big), \end{aligned} \tag{15}$$

where $p_\theta(t|x,a)$ is the prior distribution for generating latent trees, $q_\phi(t|x,y,a)$ is the corresponding posterior distribution, $\log P_\theta(y|x,a,t)$ is the log-likelihood function by assuming

that the latent tree t is already known, and $\mathbb{E}_{q_\phi(t|x,y,a)}[\log P_\theta(y|x,a,t)]$ is the expected log-likelihood function over $q_\phi(t|x,y,a)$ by considering all the potential trees. The KL term acts as a regularizer to force the matching of the prior and the posterior distributions. During training, $q_\phi(t|x,y,a)$ is used to induce the tree. For inference, $p_\theta(t|x,a)$ is used since y is still unknown.

In practice, a scale hyper-parameter β can be used to control the behaviour of the KL term (Bowman et al., 2016b),

$$\begin{aligned}\mathcal{L}_{\text{ELBO}} = &- \mathbb{E}_{q_\phi(t|x,y,a)}[\log P_\theta(y|x,a,t)] \\ &+ \beta \text{KL}\Big(q_\phi(t|x,y,a)||p_\theta(t|x,a)\Big).\end{aligned} \tag{16}$$

The first term is an **expectation term** and the second term is a **KL term**. Eq 16 is a standard VAE model for the ABSA task, which, however, has not been discussed in the research literature. It can be trained using the tree entropy (Kim et al., 2019b) and neural mutual information estimation (Fang et al., 2019). However, both are slow because they both need to consider a large batch of tree samples. To model $q_\phi(t|x,y,a)$, we instead calculate a score function $\mathbf{s}^q$ for the posterior by a MLP layer similar to Eq 1,

$$\mathbf{s}^q = \text{softmax}\Big(\mathbf{u}_q \sigma(\mathbf{W}_q \mathbf{H}' + \mathbf{W}_{a,q}\mathbf{h}'_a)\Big), \tag{17}$$

where $\mathbf{u}_q$, $\mathbf{W}_q$ and $\mathbf{W}_{a,q}$ are parameters, $\mathbf{H}'$ and $\mathbf{h}'_a$ are the posterior sentence and aspect representations respectively given y. To ensure that y can guide the encoder, we feed the input sequence together with y to BERT by using "[CLS] w_1 w_2 $\ldots w_n$ [SEP] w_f w_{f+1} $\ldots w_e$ y" to obtain $\mathbf{H}'$.

3.2 Correlation with Our Model

Our method can be regarded as a novel simplification to the above model, which can be shown by correlating the expectation term and the KL term defined in Eq 16 with the attention scores in Eq 1 and Eq 4, respectively. In particular, we consider converting t into a special type of tree distance, namely the aspect-to-context attention scores. Then we delegate the probability distribution over structured tree samples to a set of attention scores. Intuitively, if the attention scores are similar, the generated trees should be highly similar.

Approximate Expectation Term Considering the gradient of the first expectation term with respect to ϕ is,

$$\begin{aligned}&\frac{\partial \mathbb{E}_{q_\phi(t|x,y,a)}[\log P_\theta(y|x,a,t)]}{\partial \phi} \\ &= \mathbb{E}_{q_\phi(t|x,y,a)}[\log P_\theta(y|x,a,t) \frac{\partial \log q_\phi(t|x,y,a)}{\partial \phi}].\end{aligned} \tag{18}$$

Assuming that the posterior $q_\phi(t|x,y,a)$ is approximate to $Q_\phi(t|x,a)$ given by the recognition network, Eq 18 is equivalent to $\mathcal{L}_{\text{rl}}$ in Eq 11.

Approximate KL Term The KL term resembles $\mathcal{L}_{att}$ in Eq 12 for $\beta = \lambda_{att}$, namely $\text{KL}\Big(q_\phi(t|x,y,a)||p_\theta(t|x,a)\Big) \approx \text{KL}(\boldsymbol{\alpha}, \mathbf{s}^p)$. First, we delegate the probability distribution over tree samples to a set of attention scores. In particular, we use $\mathbf{s}^p$ and $\mathbf{s}^q$ as the proxies for $p_\theta(t|x,a)$ and $q_\phi(t|x,y,a)$, respectively. This is equivalent to say that the posterior scores $\mathbf{s}^q$ and the prior score $\mathbf{s}^p$ are fed to Algorithm 1 to derive the corresponding trees during training. Second, since both $\mathbf{s}^q$ and the attention score $\boldsymbol{\alpha}$ in Eq 4 are directly supervised by the output label y, we can safely assume that $\mathbf{s}^q \approx \boldsymbol{\alpha}$. Then the KL term $\text{KL}(\mathbf{s}^q, \mathbf{s}^p)$ in Eq 16 becomes $\text{KL}(\boldsymbol{\alpha}, \mathbf{s}^p)$, which is the attention-based regularization loss defined in Eq 12.

4 Experiments

We perform experiments on eight aspect-based sentiment analysis benchmarks, including six English datasets, one Chinese dataset, and one Korean datase. The data statistics is shown in Appendix A.3. We use Stanza (Qi et al., 2020) as the external parser to produce dependency parses for comparing with dependency tree based models, reporting accuracy (Acc.) and macro-f1 (F1) scores for each model. More details are presented in Appendix A.1.

MAMS Jiang et al. (2019) provide a recent challenge dataset with 4,297 sentences and 11,186 aspects. We take it as the main dataset because it is a large-scale multi-aspect dataset with more aspects in each sentence compared to the other datasets. MAMS-small is a small version of MAMS.

Chinese hotel reviews dataset Liu et al. (2020) provide manually annotated 6,339 targets and 2,071 items for multi-target sentiment analysis.

Korean automotive comments dataset Hyun et al. (2020) provide a dataset with 30,032 comment-aspect pairs in Korean.

SemEval datasets We use five SemEval datasets, including twitter posts (`Twitter`) from Dong et al. (2014), laptop comments (`Laptop`) provided

Model	Acc	F1
BERT-SPC	84.08	83.52
depGCN	83.11	82.42
depGCN + $\mathcal{L}_{\text{td}}$	83.41	82.78
kumaGCN	83.86	83.20
kumaGCN + $\mathcal{L}_{\text{td}}$	84.08	83.55
viGCN	83.93	83.39
dotGCN	84.53	83.97
- $\mathcal{L}_{\text{td}}$	84.46 (-0.07)	83.85 (-0.12)
- $\mathcal{L}_{\text{rl}}$	83.48 (–1.05)	84.01 (+0.04)
- $\mathcal{L}_{\text{att}}$	84.01 (-0.52)	83.40 (-0.57)

Table 1: Development results on MAMS dev set. All models are based on BERT.

by Pontiki et al. (2014), restaurant reviews of SemEval 2014 task 4 (`Rest14`; Pontiki et al. 2014), SemEval 2015 task 12 (`Rest15`; Pontiki et al. 2015) and SemEval 2016 task 5 (`Rest16`; Pontiki et al. 2016). These datasets are pre-processed following Tang et al. (2016) and Zhang et al. (2019).

4.1 Baselines

We denote our model as `dotGCN` (**d**iscrete **o**pinion **t**ree GCN), making comparisons with BERT-based models, including models without using trees and dependency tree based models. In addition, the variational inference baseline (Section 3.1) is denoted as **viGCN**. Baselines are (1) **BERT-SPC** is a simple baseline by fine-tuning the vector of "[CLS]" of BERT from Jiang et al. (2019); (2) **AEN**. Song et al. (2019) use an attentional encoder with BERT; (3) **CapsNet**. Jiang et al. (2019) combine capsule network with BERT; (4) **Hard-Span**. Hu et al. (2019) use RL to determine aspect-specific opinion spans; (5) **depGCN**. Zhang et al. (2019) applies aspect-specific GCNs over dependency trees; (6) **RGAT**. Wang et al. (2020) use relational graph attention networks over aspect-centered dependency trees to incorporate the dependency edge type information; (7) **SAGAT**. Huang et al. (2020) use graph attention network and BERT, exploring both syntax and semantic information in the sequence; (8) **DGEDT**. Tang et al. (2020) jointly consider BERT outputs and dependency tree based representations by a bidirectional GCN. (9) **kumaGCN**. Chen et al. (2020) combine the dependency trees and latent graphs induced by self-attention neural networks;

4.2 Development Results

We perform development experiments using MAMS since this is the largest dataset and the examples are more challenging compared to the other datasets. We implement three baselines, including BERT-SPC, depGCN and kumaGCN. For fair comparison, we also combine depGCN and kumaGCN with the syntax regularization loss in Eq 7 by calculating syntactic distances on the input dependency trees with respect to the aspect terms.

Method	**MAMS**		**Small**		**Multilingual**	
	Acc	F1	Acc	F1	Ch-F1	Ko-F1
BERT-SPC	82.22	-	79.44	-	-	-
CapsNet	83.39	-	80.91	-	-	-
CapsNet-DR	82.97	-	80.09	-	-	-
BERT-SPC*	83.01	82.76	80.91	80.39	80.92	61.17
depGCN + $\mathcal{L}_{\text{td}}^{*}$	84.36	83.88	81.59	80.81	NA	NA
kumaGCN + $\mathcal{L}_{\text{td}}^{*}$	84.37	83.83	81.59	81.10	NA	NA
dotGCN	**84.95**	**84.44**	**82.34**	**81.73**	**81.53**	**62.78**

Table 2: Results on two MAMS datasets and the multilingual review datasets. $*$ denotes our implementation.

Table 1 shows the results on MAMS validation-set. BERT-SPC achieves 84.08 accuracy and 83.52 F1. Surprisingly, the dependency tree based models cannot outperform BERT-SPC, which verifies the limitation of using cross-domain dependency parsers for this task. kumaGCN outperforms depGCN due to its ability to include an implicit latent graph. Adding the syntax regularization loss generally improves the model performance of syntax-based models. In particular, kumaGCN + $\mathcal{L}_{\text{td}}$ is on par with BERT-SPC.

viGCN outperforms kumaGCN + $\mathcal{L}_{\text{td}}$ and depGCN + $\mathcal{L}_{\text{td}}$, which shows the potential of structured latent tree models. Our dotGCN model achieves 84.53 accuracy and 83.97 F1, outperforming all the baselines by a large margin, which empirically shows the induced discrete opinion tree is promising to this task. Compared to viGCN, our model gives better scores. In addition, our model converges nearly 1.8 times faster (0.66h/epoch v.s. 1.25h/epoch) than viGCN. dotGCN does not have to calculate the true posterior distribution over structured tree samples and thus largely reduce computation overhead.

Ablation Study Table 1 shows ablation studies on MAMS validation set by removing three proposed loss items during training, namely $\mathcal{L}_{\text{td}}$, $\mathcal{L}_{\text{rl}}$ and $\mathcal{L}_{\text{att}}$. We can observe that the model performance degrades after removing either one of them. Removing the syntax regularization loss $\mathcal{L}_{\text{td}}$ slightly hurts the performance. Without using the attention consistency loss $\mathcal{L}_{\text{att}}$, the model falls behind BERT-SPC, which suggests the importance of our proposed attention consistency regularizations. Excluding the reinforcement learning loss leads to the

Model	Twitter		Laptop		Rest14		Rest15		Rest16		Average	
	Acc	F1	Acc	F1	Acc	F1	Acc	F1	Acc	F1	Acc	F1
AEN	75.14	74.15	76.96	73.67	84.29	77.22	-	-	-	-	-	-
RGAT	76.15	74.88	78.21	74.07	**86.60**	**81.35**	-	-	-	-	-	-
BERT-SPC*	73.41	72.38	80.56	77.20	84.55	75.74	83.03	63.92	90.75	74.00	82.46	72.65
depGCN*	75.58	74.58	**81.19**	77.67	85.00	78.79	84.13	67.28	91.39	74.25	83.46	74.51
SAGAT	75.40	74.17	80.37	76.94	85.08	77.94	-	-	-	-	-	-
DGEDT	77.90	75.40	79.80	75.60	86.30	80.00	84.00	71.00	91.90	79.00	83.98	76.2
depGCN + $\mathcal{L}^*_{\text{td}}$	75.49	76.73	79.31	75.84	86.43	80.72	84.69	70.89	92.37	79.40	83.66	76.72
dotGCN	**78.11**	**77.00**	81.03	**78.10**	86.16	80.49	**85.24**	**72.74**	**93.18**	**82.32**	**84.74**	**78.13**

Table 3: Results on five SemEval datasets. All the models are based on BERT. * denotes our implementations.

biggest performance drop (Acc: 84.53 → 83.48) among the three settings. This shows that the reinforcement learning component plays a central role in the full model.

4.3 Main Results

MAMS Table 2 shows the results of dotGCN and the baselines from Jiang et al. (2019) on the MAMS test set. We implement BERT-SPC, denoted as BERT-SPC*, which outperforms the BERT-SPC model of Jiang et al. (2019). Compared to baselines (BERT-SPC, CapsNet, CapsNet-DR and BERT-SPC*) without using dependency trees, dotGCN gives significantly better results ($p < 0.01$). For fair comparison with dependency tree based models, we also implement depGCN+$\mathcal{L}^*_{\text{td}}$ and kumaGCN+$\mathcal{L}^*_{\text{td}}$. depGCN+$\mathcal{L}^*_{\text{td}}$ achieves 84.36 accuracy and 83.88 F1 on the MAMS test set. kumaGCN+$\mathcal{L}^*_{\text{td}}$ gives similar results with 84.37 accuracy and 83.83 F1. Our dotGCN outperforms all the baselines, giving 84.95 accuracy and 84.44 F1. In terms of the averaged accuracy of F1 scores on MAMS and MAMS-small, dotGCN is significantly better than depGCN and kumaGCN ($p < 0.05$). The results demonstrate that the induced aspect-specific discrete opinion trees are promising to handle multi-aspect sentiment tasks.

Multilingual The results[3] on the Chinese hotel review dataset are shown in Table 2. dotGCN outperforms the baseline BERT-SPC* by 0.72 accuracy points and 0.61 F1, respectively. The result shows that our model can be generalized across languages without relying on language-specific parsers. On the Korean dataset, we obtain 5.20 accuracy and 11.61 F1 improvements compared to the LCF-BERT (Zeng et al., 2019), which is the best BERT-based model. These results show that our model can be well generalized to multiple languages and may potentially benefits low-resource languages for this task.

Model	Laptops	Restaurants
BERT-SPC	74.57	82.66
Soft-Span	74.92	82.68
Hard-Span	74.10	83.91
dotGCN	**76.65**	**84.11**

Table 4: Comparisons with span-based RL.

SemEval Table 3 shows the results of our model on the SemEval datasets. First, tree based graph neural network models are generally better than BERT-SPC. On the five datasets, which are relatively small, our model still achieves competitive performances in terms of the averaged F1 and accuracy scores as shown in Table 3. In particular, our model in general outperforms depGCN and depGCN + $\mathcal{L}^*_{\text{td}}$ on four out of five datasets, which verifies that the reinforced discrete opinion trees can be promising structured representations compared to auto-parsed dependency trees.

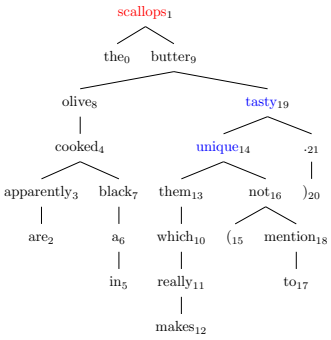
We also compare our models with span-based reinforcement learning models (Hard-Span; Hu et al. (2019)) on the dataset of laptops and restaurants preprocessed by Tay et al. (2018). As shown in Table 4, our model outperforms Hard-Span by 2.55 accuracy points on laptops[4]. On restaurants, our model achieves a comparable result to Hard-Span. It shows that the opinion tree is a better representation compared to an opinion span.

4.4 Case Study

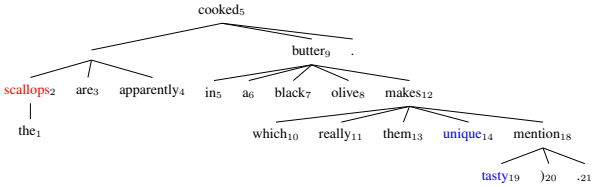
Figure 3a and Figure 3b show the induced tree and dependency parse for the aspect term "scallops", respectively. The opinion words "unique" and "tasty"

[3]Since the Hotel dataset is based on Chinese characters, there are no annotated words. To avoid character-level dependency parsing, we omit them in Table 2 for consistency.

[4]Since the code of span-based RL methods is not publicly available, we do not include a significant test here.



(a) An induced tree for long-term dependencies.



(b) The corresponding dependency tree by Stanza.

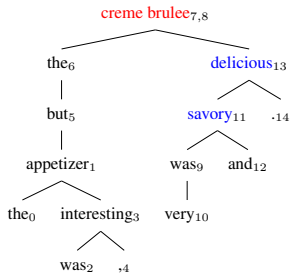
Figure 3: Tree examples. *red*: aspect, *blue*: opinions, *number*: word index. The sentence is "the **scallops** are apparently cooked in a black olive butter which really makes them unique (not to mention tasty) ."

are far away from the aspect (more than 10 words) in the dependency tree. In the induced tree by dotGCN, the opinion word "tasty" and "unique" are 2 and 3 depths from the aspect "scallops" respectively, which shows that dotGCN can potentially handle complex interactions among aspects and opinion contexts. In addition, the tree induced by dotGCN is binarized, and the root node can contain multiple words as shown in Figure 4a.

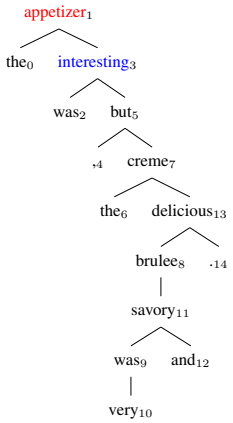
Figure 4a and Figure 4b show the induced trees for two aspect terms with different sentiment polarities. For "creme brulee", the policy network assigns high weights to both "delicious" and "savory". Interestingly, it assigns a higher weight to "delicious" than "savory", though "savory" is closer to its aspect term than "delicious". For "appetizer", the word "interesting" receives higher attention scores than the other two sentiment words. These results show that dotGCN is able to distinguish different sentiment contexts for different aspect terms in the same sentence.

4.5 Analysis

Distances between Aspect Terms and Opinion Words Figure 5 shows the distances between aspect terms and opinion words. We use the annotated opinion words of Rest16 provided by Fan et al. (2019) to compare our induced trees and dependency trees. The distances calculated over the original sequences are also included. We can observe



(a) An induced tree for "creme brulee".



(b) An induced tree for "appetizer".

Figure 4: Tree examples. The sentence is "the **appetizer** was interesting , but the **creme brulee** was very savory and delicious ."

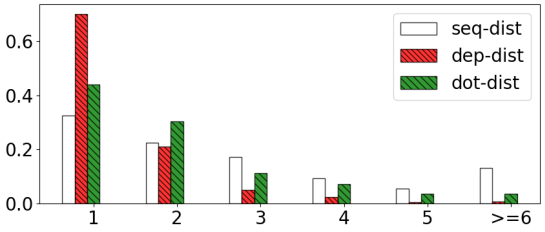


Figure 5: Distances between aspect terms and opinion words. *seq*: sequential structure; *dep*: dependency tree, *dot* denotes our discrete opinion tree.

that the distance distribution over the sequences is relatively flat compared to that over tree structures. For the two tree structures, nearly 90% of opinion words are within 3 depths from the aspect terms. The distance distribution of our induced trees is similar to that of the dependency trees, which empirically demonstrates that induced discrete trees are able to capture the interactions between aspect terms and opinions. By treating dependency trees as gold standard, our tree inducer obtains 35.4% unlabeled attachment scores (UAS), which shows the induced trees are significantly different from the dependency trees although both can connect opinion words with aspect terms.

Low frequent aspects Table 5 shows the classifi-

Frequency	depGCN+$\mathcal{L}_{td}^{*}$	dotGCN
0	81.96	**83.53** (+1.57)
1	74.63	74.63
>=2	85.60	86.29

Table 5: Classification accuracy of test set with respect to the frequency of aspects in training set using MAMS.

cation accuracy of the MAMS test set with respect to the aspect frequency. For aspect terms which appear in the training corpus, both methods give similar results. However, for unseen aspects, dotGCN gives better results than depGCN. This is potentially due to the severe parsing errors for the low-frequent aspects. dotGCN does not depend on external parsers and thus can circumvent this problem. It empirically suggests that the induced tree structures have strong robustness for capturing the aspect-opinion interactions compared to depGCN.

5 Related Work

Tree Induction for ABSA There has been much work on unsupervised discrete induction (Bowman et al., 2016a; Shen et al., 2018b; Kim et al., 2019b,a; Jin et al., 2019; Cao et al., 2020; Yang et al., 2021; Dai et al., 2021), which aims to obtain general constituent trees without explicit syntax annotations and task-dependent supervised signals. We focus on learning task-specific tree-structures for ABSA, where the tree is fully binarized and lexicalized. Choi et al. (2018) propose Gumbel Tree-LSTM for learning task-specific tree for semantic compositions. Similarly, Maillard et al. (2019) propose an unsupervised chart parser for jointly learning sentence embeddings and syntax. However, they focus on sentence-level tasks and do not consider aspect information.

Aspect-level Sentiment Classification Much recent work has explored neural attention mechanism to this task (Tang et al., 2016; Ma et al., 2017; Li et al., 2018; Liang et al., 2019). Among tree-based methods, Zhang et al. (2019) and Sun et al. (2019b) encode dependency tree using GCN for aspect-level sentiment analysis; Zhao et al. (2019) use GCN to model fully connected graphs between aspect terms; Wang et al. (2020) use relational graph attention networks to incorporate the dependency edge type information, and construct aspect-specific graph structures; Barnes et al. (2021) attempt to directly predict dependency-based sentiment graphs. Tang et al. (2020) use duel-transformer structure to enhance the dependency graph for this task. Our work is similar in that we also consider the structure dependencies, but different in that we rely on automatically induced tree structures instead of external parses. Chen et al. (2020) propose to induce aspect-specific latent graph by sampling from self-attention-based Hard Kumaraswamy distributions (Bastings et al.). However, to achieve competitive performance, their method still requires a combination of external dependency parse trees and the induced latent graphs.

Sun et al. (2019a) and Xu et al. (2019) constructed aspect related auxiliary sentences as inputs to BERT (Devlin et al., 2019) for strong contextual encoders. Xu et al. (2019) proposed BERT-based post training for enhancing domain-specific contextual representations for aspect sentiment analysis. Our work shares a similar feature extraction approach, but differently we focus on inducing latent trees for ABSA.

6 Conclusion

We proposed a method to induce aspect-specific discrete opinion trees for aspect-based sentiment analysis, obtaining trees by viewing aspect-to-context attention scores as syntactic distances. The attention scores are trained using both RL and a novel attention-based regularization. Our model empirically achieves competitive performance compared with dependency tree based models, while being independent of parsers. We also provide a theoretic view of our method using variational inference.

Acknowledgements

Zhiyang Teng and Yue Zhang are the corresponding authors. Our thanks to anonymous reviewers for their insightful comments and suggestions. We appreciate Prof. Pengyuan Liu sharing the Chinese Hotel dataset, Prof. Jingjing Wang sharing the reinforcement learning code of Wang et al. (2019), Mr. Chuang Fan helping obtain the MAMS-Small dataset, Prof. Hwanjo Yu and Mr. Dongmin Hyun sharing the Korean automotive datasets, Prof. Dejiang Dou and Mr. Amir Veyseh responding to our questions when reproducing their results on MAMS, Mr. Zhen Wu for releasing their codes of Wu et al. (2020) upon our request. We thank Dr. Xuebin Wang for providing us with 2 V100 GPU cards for use. This publication is conducted with the financial support of "Pioneer" and "Leading Goose" R&D Program of Zhejiang under Grant Number 2022SDXHDX0003.

References

Jeremy Barnes, Robin Kurtz, Stephan Oepen, Lilja Øvrelid, and Erik Velldal. 2021. Structured sentiment analysis as dependency graph parsing. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 3387–3402, Online. Association for Computational Linguistics.

Joost Bastings, Wilker Aziz, and Ivan Titov. Interpretable neural predictions with differentiable binary variables.

Samuel R. Bowman, Jon Gauthier, Abhinav Rastogi, Raghav Gupta, Christopher D. Manning, and Christopher Potts. 2016a. A fast unified model for parsing and sentence understanding. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1466–1477, Berlin, Germany. Association for Computational Linguistics.

Samuel R. Bowman, Luke Vilnis, Oriol Vinyals, Andrew Dai, Rafal Jozefowicz, and Samy Bengio. 2016b. Generating sentences from a continuous space. In *Proceedings of The 20th SIGNLL Conference on Computational Natural Language Learning*, pages 10–21, Berlin, Germany. Association for Computational Linguistics.

Steven Cao, Nikita Kitaev, and Dan Klein. 2020. Unsupervised parsing via constituency tests. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4798–4808, Online. Association for Computational Linguistics.

Chenhua Chen, Zhiyang Teng, and Yue Zhang. 2020. Inducing target-specific latent structures for aspect sentiment classification. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5596–5607, Online. Association for Computational Linguistics.

Jihun Choi, Kang Min Yoo, and Sang goo Lee. 2018. Learning to compose task-specific tree structures. In *AAAI*.

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*, pages 1816–1829, Online. Association for Computational Linguistics.

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, Minneapolis, Minnesota. 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 *Proc. of ACL*.

Chunning Du, Haifeng Sun, Jingyu Wang, Qi Qi, Jianxin Liao, Tong Xu, and Ming Liu. 2019. Capsule network with interactive attention for aspect-level sentiment classification. 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 5489–5498, Hong Kong, China. Association for Computational Linguistics.

Long Duong, Trevor Cohn, Steven Bird, and Paul Cook. 2015. A neural network model for low-resource universal dependency parsing.

Zhifang Fan, Zhen Wu, Xin-Yu Dai, Shujian Huang, and Jiajun Chen. 2019. Target-oriented opinion words extraction with target-fused neural sequence labeling. 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 2509–2518, Minneapolis, Minnesota. Association for Computational Linguistics.

Le Fang, Chunyuan Li, Jianfeng Gao, Wen Dong, and Changyou Chen. 2019. Implicit deep latent variable models for text generation. 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 3946–3956, Hong Kong, China. Association for Computational Linguistics.

Mengting Hu, Shiwan Zhao, Honglei Guo, Renhong Cheng, and Zhong Su. 2019. Learning to detect opinion snippet for aspect-based sentiment analysis. *CoRR*, abs/1909.11297.

Lianzhe Huang, Xin Sun, Sujian Li, Linhao Zhang, and Houfeng Wang. 2020. Syntax-aware graph attention network for aspect-level sentiment classification. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 799–810, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Dongmin Hyun, Junsu Cho, and Hwanjo Yu. 2020. Building large-scale English and Korean datasets for aspect-level sentiment analysis in automotive domain. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 961–966, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Long Jiang, Mo Yu, Ming Zhou, Xiaohua Liu, and Tiejun Zhao. 2011. Target-dependent Twitter sentiment classification. In *Proceedings of the 49th Annual Meeting of the Association for Computational*

Linguistics: Human Language Technologies, pages 151–160, Portland, Oregon, USA. 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 *EMNLP-IJCNLP*, pages 6281–6286.

Lifeng Jin, Finale Doshi-Velez, Timothy Miller, Lane Schwartz, and William Schuler. 2019. Unsupervised learning of PCFGs with normalizing flow. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2442–2452, Florence, Italy. Association for Computational Linguistics.

Yoon Kim, Chris Dyer, and Alexander Rush. 2019a. Compound probabilistic context-free grammars for grammar induction. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2369–2385, Florence, Italy. Association for Computational Linguistics.

Yoon Kim, Alexander Rush, Lei Yu, Adhiguna Kuncoro, Chris Dyer, and Gábor Melis. 2019b. Unsupervised recurrent neural network grammars. 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 1105–1117, Minneapolis, Minnesota. Association for Computational Linguistics.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Diederik P. Kingma and Max Welling. 2014. Auto-encoding variational bayes. *Stat*, 1050:1.

Thomas N. Kipf and Max Welling. 2017. Semi-supervised classification with graph convolutional networks. In *Proc. of ICLR*.

Kemal Kurniawan, Lea Frermann, Philip Schulz, and Trevor Cohn. 2021. PPT: parsimonious parser transfer for unsupervised cross-lingual adaptation. *CoRR*, abs/2101.11216.

Xin Li, Lidong Bing, Wai Lam, and Bei Shi. 2018. Transformation networks for target-oriented sentiment classification. In *Proc. of ACL*, pages 946–956, Australia. Association for Computational Linguistics.

Yunlong Liang, Fandong Meng, Jinchao Zhang, Jinan Xu, Yufeng Chen, and Jie Zhou. 2019. A novel aspect-guided deep transition model for aspect based sentiment analysis. In *Proc. of EMNLP-IJCNLP*, pages 5568–5579, HK, China.

Pengyuan Liu, Yongsheng Tian, Chengyu Du, and Likun Qiu. 2020. Construction and analysis of chinese multi-target sentiment classification dataset. In *Proceedings of the 19th Chinese National Conference on Computational Linguistics*, pages 604–615, China.

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*, pages 4068–4074.

Jean Maillard, Stephen Clark, and Dani Yogatama. 2019. Jointly learning sentence embeddings and syntax with unsupervised tree-lstms. *Natural Language Engineering*, 25(4):433–449.

Thien Hai Nguyen and Kiyoaki Shirai. 2015. PhraseRNN: Phrase recursive neural network for aspect-based sentiment analysis. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 2509–2514, Lisbon, Portugal. Association for Computational Linguistics.

Maria Pontiki, Dimitris Galanis, Haris Papageorgiou, Ion Androutsopoulos, Suresh Manandhar, Mohammad AL-Smadi, Mahmoud Al-Ayyoub, Yanyan Zhao, Bing Qin, Orphee De Clercq, Veronique Hoste, Marianna Apidianaki, Xavier Tannier, Natalia Loukachevitch, Evgeniy Kotelnikov, Núria Bel, Salud María Jiménez-Zafra, and Gülşen Eryiğit. 2016. Semeval-2016 task 5 : aspect based sentiment analysis. In *Proc. of the Workshop on SemEval*.

Maria Pontiki, Dimitris Galanis, Haris Papageorgiou, Suresh Manandhar, and Ion Androutsopoulos. 2015. Semeval-2015 task 12: Aspect based sentiment analysis. In *Proc. of the Workshop on SemEval*.

Maria Pontiki, Dimitris Galanis, John Pavlopoulos, Harris Papageorgiou, Ion Androutsopoulos, and Suresh Manandhar. 2014. Semeval-2014 task 4: Aspect based sentiment analysis. In *Proc. of the Workshop on SemEval*.

Amir Pouran Ben Veyseh, Nasim Nouri, Franck Dernoncourt, Dejing Dou, and Thien Huu Nguyen. 2020. Introducing syntactic structures into target opinion word extraction with deep learning. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 8947–8956, Online. Association for Computational Linguistics.

Peng Qi, Yuhao Zhang, Yuhui Zhang, Jason Bolton, and Christopher D. Manning. 2020. Stanza: A python natural language processing toolkit for many human languages. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 101–108, Online. Association for Computational Linguistics.

Guy Rotman and Roi Reichart. 2019. Deep Contextualized Self-training for Low Resource Dependency Parsing. *Transactions of the Association for Computational Linguistics*, 7:695–713.

Reuven Y. Rubinstein and Dirk P. Kroese. 2016. *Simulation and the Monte Carlo method*, volume 10. John Wiley & Sons.

Ronald Seoh, Ian Birle, Mrinal Tak, Haw-Shiuan Chang, Brian Pinette, and Alfred Hough. 2021. Open aspect target sentiment classification with natural language prompts. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 6311–6322, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Yikang Shen, Zhouhan Lin, Athul Paul Jacob, Alessandro Sordoni, Aaron Courville, and Yoshua Bengio. 2018a. Straight to the tree: Constituency parsing with neural syntactic distance. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1171–1180, Melbourne, Australia. Association for Computational Linguistics.

Yikang Shen, Zhouhan Lin, Chin wei Huang, and Aaron Courville. 2018b. Neural language modeling by jointly learning syntax and lexicon. In *International Conference on Learning Representations*.

Jiaxin Shi, Lei Hou, Juanzi Li Li, Zhiyuan Liu, and Hanwang Zhang. 2019. Learning to embed sentences using attentive recursive trees. In *AAAI*.

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

Chi Sun, Luyao Huang, and Xipeng Qiu. 2019a. Utilizing BERT for aspect-based sentiment analysis via constructing auxiliary sentence. In *Proc. of NAACL-HLT*, pages 380–385, USA.

Kai Sun, Richong Zhang, Samuel Mensah, Yongyi Mao, and Xudong Liu. 2019b. Aspect-level sentiment analysis via convolution over dependency tree. In *Proc. of EMNLP-IJCNLP*, pages 5678–5687, HK, China.

Duyu Tang, Bing Qin, and Ting Liu. 2016. Aspect level sentiment classification with deep memory network. In *Proc. of EMNLP*, pages 214–224, USA.

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.

Yi Tay, Luu Anh Tuan, and Siu Cheung Hui. 2018. Learning to attend via word-aspect associative fusion for aspect-based sentiment analysis. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32.

Clara Vania, Yova Kementchedjhieva, Anders Søgaard, and Adam Lopez. 2019. A systematic comparison of methods for low-resource dependency parsing on genuinely low-resource languages. 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 1105–1116, Hong Kong, China. Association for Computational Linguistics.

Jingjing Wang, Changlong Sun, Shoushan Li, Jiancheng Wang, Luo Si, Min Zhang, Xiaozhong Liu, and Guodong Zhou. 2019. Human-like decision making: Document-level aspect sentiment classification via hierarchical reinforcement learning. 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 5581–5590, Hong Kong, China. Association for Computational Linguistics.

Kai Wang, Weizhou Shen, Yunyi Yang, Xiaojun Quan, and Rui Wang. 2020. Relational graph attention network for aspect-based sentiment analysis. *arXiv preprint arXiv:2004.12362*.

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, Austin, Texas. Association for Computational Linguistics.

Ronald J Williams. 1992. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning*, 8(3-4):229–256.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Zhen Wu, Chengcan Ying, Xinyu Dai, Shujian Huang, and Jiajun Chen. 2020. Transformer-based multi-aspect modeling for multi-aspect multi-sentiment analysis. In *Natural Language Processing and Chinese Computing*, pages 546–557, Cham. Springer International Publishing.

Zeguan Xiao, Jiarun Wu, Qingliang Chen, and Congjian Deng. 2021. BERT4GCN: Using BERT intermediate layers to augment GCN for aspect-based sentiment classification. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 9193–9200, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

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

Songlin Yang, Yanpeng Zhao, and Kewei Tu. 2021. Neural bi-lexicalized PCFG induction. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 2688–2699, Online. Association for Computational Linguistics.

Pengcheng Yin, Chunting Zhou, Junxian He, and Graham Neubig. 2018. StructVAE: Tree-structured latent variable models for semi-supervised semantic parsing. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 754–765, Melbourne, Australia. Association for Computational Linguistics.

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).

Chen Zhang, Qiuchi Li, and Dawei Song. 2019. Aspect-based sentiment classification with aspect-specific graph convolutional networks. In *Proc. of EMNLP-IJCNLP*, pages 4567–4577, HK, China.

Pinlong Zhao, Linlin Hou, and Ou Wu. 2019. Modeling sentiment dependencies with graph convolutional networks for aspect-level sentiment classification. In *arXiv:1906.04501*.

Dataset		#Pos.	#Neg.	#Neu.	Total
Laptops	Train	767	673	373	1811
	Dev	220	193	87	500
	Test	341	128	169	638
Restaurants	Train	685	1,886	531	3,120
	Dev	278	120	102	500
	Test	728	196	196	1,120

Table 6: Statistics of the dataset of laptops and restaurants preprocessed by Tay et al. (2018).

A Appendix

A.1 Settings

Our codes are implemented based on the PyTorch Transformers library (Wolf et al., 2020). We use *bert-based-uncased*[5] for English, *bert-base-chinese*[6] for Chinese, *bert-base-multilingual-uncased*[7] for Korean. We tune the hyper-parameters on the MAMS dataset. We select the best model according to the accuracy scores on the development set. For each model, we train it 10 epochs with the Adam optimizer (Kingma and Ba, 2014). The initial learning rate for fine-tuning BERT parameters is $2e^{-5}$ and the weight decay is $1e^{-5}$. The number of GCN layers is 2 by following Zhang et al. (2019). The hidden size of the MLP layer in Eq 1 is 256. For the policy network training, we generate $M = 3$ trees. $\lambda_{rl} = \lambda_{att} = \lambda_{sd} = 0.1$. For the variational inference model, $\beta = 0.05$. We try five options for these hyper-parameters ($\lambda_{rl}, \lambda_{att}, \lambda_{sd}$) including 0, 0.01, 0.05, 0.1 and 0.2. We report accuracy (Acc.) and macro-f1 (F1) scores for each model. For the other settings about neural network architectures and reinforcement learning, we follow Zhang et al. (2019) and Shi et al. (2019), respectively.

We run our models using a single GPU Card (TitanXP 1080ti or Titan XP 2080 or V100). Each training epoch for MAMS taskes about 40 mins.

A.2 Statistics of Tay et al. (2018)'s dataset

We compare our discrete opinion tree RL model with span-based RL model on a dataset preprocessed by Tay et al. (2018). Table 6 shows the statistics.

[5] https://storage.googleapis.com/bert_models/2020_02_20/uncased_L-12_H-768_A-12.zip

[6] https://storage.googleapis.com/bert_models/2018_11_03/chinese_L-12_H-768_A-12.zip

[7] https://huggingface.co/bert-base-multilingual-uncased

Dataset		#Pos.	#Neu.	#Neg.
TWITTER	Train/Test	1,561/173	3,127/346	1,560/173
LAPTOP	Train/Test	994/341	464/169	870/128
REST14	Train/Test	2,164/728	637/196	807/196
REST15	Train/Test	912/326	36/34	256/182
REST16	Train/Test	1,240/469	69/30	439/117
CHINESE-HOTEL	Train/Test	2,250/751	383/128	2,120/707
MAMS	Train/Dev/Test	3,380/403/400	5,042/604/607	2,764/325/329
MAMS-SMALL	Train/Dev/Test	1,089/403/400	1,627/604/607	892/325/329
KOREAN-AUTO	Train/Test	4,787/1,180	14,212 /3,583	5,027/1,243

Table 7: Dataset Statistics.

A.3 Data

Table 7 shows the data statistics. The MAMS dataset can be obtain from https://github.com/siat-nlp/MAMS-for-ABSA. The five SemEval datasets can be downloaded from https://github.com/GeneZC/ASGCN/tree/master/datasets, the Chinese dataset can be obtained from https://github.com/NLPBLCU/ and the Korean dataset can be obtained from https://github.com/dmhyun/alsadata.