Does syntax matter? A strong baseline for Aspect-based Sentiment Analysis with RoBERTa

Junqi Dai[1,*], Hang Yan[1,*], Tianxiang Sun[1], Pengfei Liu[2], Xipeng Qiu[1,†]
[1]Shanghai Key Laboratory of Intelligent Information Processing, Fudan University
[1]School of Computer Science, Fudan University
[2]Carnegie Mellon University
{jqdai19,hyan19,txsun19,xpqiu}@fudan.edu.cn
pliu3@cs.cmu.edu

Abstract

Aspect-**B**ased **S**entiment **A**nalysis (**ABSA**), aiming at predicting the polarities for aspects, is a fine-grained task in the field of sentiment analysis. Previous work showed syntactic information, e.g. dependency trees, can effectively improve the ABSA performance. Recently, pre-trained models (PTMs) also have shown their effectiveness on ABSA. Therefore, the question naturally arises whether PTMs contain sufficient syntactic information for ABSA so that we can obtain a good ABSA model only based on PTMs. In this paper, we firstly compare the induced trees from PTMs and the dependency parsing trees on several popular models for the ABSA task, showing that the induced tree from fine-tuned RoBERTa (FT-RoBERTa) outperforms the parser-provided tree. The further analysis experiments reveal that the FT-RoBERTa Induced Tree is more sentiment-word-oriented and could benefit the ABSA task. The experiments also show that the pure RoBERTa-based model can outperform or approximate to the previous SOTA performances on six datasets across four languages since it implicitly incorporates the task-oriented syntactic information.[1]

1 Introduction

Aspect-based sentiment analysis (ABSA) aims to do the fine-grained sentiment analysis towards aspects (Pontiki et al., 2014, 2016). Specifically, for one or more aspects in a sentence, the task calls for detecting the sentiment polarities for all aspects. Take the sentence "great food but the service was dreadful" for example, the task is to predict the sentiments towards the underlined aspects, which expects to get polarity *positive* for aspect *food* and polarity *negative* for aspect *service*. Generally, ABSA contains aspect extraction (AE) and aspect-level sentiment classification (ALSC). We only focus on the ALSC task.

Early works of ALSC mainly rely on manually designed syntactic features, which is labor-intensive yet insufficient. In order to avoid designing hand-crafted features (Jiang et al., 2011; Kiritchenko et al., 2014), various neural network models have been proposed in ALSC (Dong et al., 2014; Vo and Zhang, 2015; Wang et al., 2016; Chen et al., 2017; He et al., 2018; Zhang et al., 2019b; Wang et al., 2020). Since the dependency tree can help the aspects find their contextual words, most of the recently proposed State-of-the-art (SOTA) ALSC models utilize the dependency tree to assist in modeling connections between aspects and their opinion words (Wang et al., 2020; Sun et al., 2019b; Zhang et al., 2019b). Generally, these dependency tree based ALSC models are implemented in three methods. The first one is to use the topological structure of the dependency tree (Dong et al., 2014; Zhang et al., 2019a; Huang and Carley, 2019; Sun et al., 2019b; Zheng et al., 2020; Tang et al., 2020); The second one is to use the tree-based distance, which counts the number of edges in a shortest path between two tokens in the dependency tree (He et al., 2018; Zhang et al., 2019b; Phan and Ogunbona, 2020); The third one is to simultaneously use both the topological structure and the tree-based distance.

Except for the dependency tree, pre-trained models (PTMs) (Qiu et al., 2020), such as BERT (Devlin et al., 2019), have also been used to enhance the performance of the ALSC task (Sun et al., 2019a; Tang et al., 2020; Phan and Ogunbona, 2020; Wang et al., 2020). From the view of interpretability of PTMs, Chen et al. (2019); Hewitt and Manning (2019); Wu et al. (2020) try to use probing methods to detect syntactic information in PTMs. Empirical results reveal that PTMs capture some kind of dependency tree structures implicitly.

[*] Equal contribution.
[†] Corresponding author.
[1] Our code will be released at https://github.com/ROGERDJQ/RoBERTaABSA.

Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pages 1816–1829
June 6–11, 2021. ©2021 Association for Computational Linguistics

Therefore, two following questions arise naturally.

Q1: Will the tree induced from PTMs achieve better performance than the tree given by a dependency parser when combined with different tree-based ALSC models? To answer this question, we choose one model from each of the three typical dependency tree based methods in ALSC, and compare their performance when combined with the parser-provided dependency tree and the off-the-shelf PTMs induced trees.

Q2: Will PTMs adapt the implicitly entailed tree structure to the ALSC task during the fine-tuning? Therefore, in this paper, we not only use the trees induced from the off-the-shelf PTMs to enhance ALSC models, but also use the trees induced from the fine-tuned PTMs (In short FT-PTMs) which are fine-tuned on the ALSC datasets. Experiments show that trees induced from FT-PTMs can help tree-based ALSC models achieve better performance than their counterparts before fine-tuning. Besides, models with trees induced from the ALSC fine-tuned RoBERTa can even outperform trees from the dependency parser.

Last but not least, we find that the base RoBERTa with an MLP layer is enough to achieve State-of-the-art (SOTA) or near SOTA performance on all six ALSC datasets across four languages, while incorporating tree structures into RoBERTa-based ALSC models does not achieve concrete improvement.

Therefore, our contributions can be summarized as:

(1) We extensively study the induced trees from PTMs and FT-PTMs. Experiments show that models using induced trees from FT-PTMs achieve better performance. Moreover, models using induced trees from fine-tuned RoBERTa outperform other trees.

(2) The analysis of the induced tree from FT-PTMs shows that it tends to be more sentiment-word-oriented, making the aspect term directly connect to its sentiment adjectives.

(3) We achieve SOTA or near SOTA performances on six ALSC datasets across four languages based on RoBERTa. We find that the RoBERTa could better adapt to ALSC and help the aspects to find the sentiment words.

2 Related Work

ALSC without Dependencies Vo and Zhang (2015) propose the early neural network model which does not rely on the dependency tree. Along this line, diverse neural network models have been proposed. Tang et al. (2016a) use the long short term memory (LSTM) network to enhance the interactions between aspects and context words. In order to model relations of aspects and their contextual words, Wang et al. (2016); Liu and Zhang (2017); Ma et al. (2017); Tay et al. (2018) incorporate the attention mechanism into the LSTM-based neural network models. Other model structures such as convolutional neural network (CNN) (Li et al., 2018; Xue and Li, 2018), gated neural network (Zhang et al., 2016; Xue and Li, 2018), memory neural network (Tang et al., 2016b; Chen et al., 2017; Wang et al., 2018), attention neural network (Tang et al., 2019) have also been applied in ALSC.
ALSC with Dependencies Early works of ALSC mainly employ traditional text classification methods focusing on machine learning algorithms and manually designed features, which took syntactic structures into consideration from the very beginning. Kiritchenko et al. (2014) combine a set of features including sentiment lexicons and parsing dependencies, from which experiments show the effectiveness of context parsing features.

A myriad of works attempt to fuse dependency tree into neural network models in ALSC. Dong et al. (2014) propose to convert the dependency tree into a binary tree first, then apply the adaptive recursive neural network to propagate information from the context words to aspects. Despite the improvement of aspect-oriented feature modeling, converting the dependency tree into a binary tree might cause syntax related words separated away from each other. In general, owing to the syntax parsing errors, early dependency tree based ALSC models do not show clear preponderance over models without the dependency tree.

However, the introduction of the neural network into the dependency parsing task enhances the parsing quality substantially (Chen and Manning, 2014; Dozat and Manning, 2017). Recent advances, leveraging graph neural network (GNN) to model the dependency tree (Zhang et al., 2019a; Huang and Carley, 2019; Sun et al., 2019b; Tang et al., 2020; Wang et al., 2020), have achieved significant performance. Among them, Zheng et al. (2020); Wang et al. (2020) attempt to convert the dependency tree into the aspect-oriented dependency tree. Instead of using the topological structure of dependency tree, He et al. (2018); Zhang et al. (2019b); Phan and Ogunbona (2020) exploit the tree-based distance between two tokens in the dependency tree.

PTMs-based Dependency Probing Over the past few years, the pre-trained models (PTMs) have dominated across various NLP tasks. Therefore, many researchers are attracted to investigate what linguistic knowledge has been captured by PTMs (Clark et al., 2019; Hewitt and Liang, 2019; Hewitt and Manning, 2019; Wu et al., 2020). Clark et al. (2019) try to use a single or a combination of head attention maps of BERT to infer the dependencies. Since BERT has many attention heads, this method can hardly fully reveal the dependency between two tokens. Hewitt and Manning (2019) propose a small learnable probing model to probe the syntax dependencies encoded in BERT. Despite very few parameters been added, it may still be very hard to tell if the syntactic information is encoded by BERT itself or by the additional parameters from the probing model. Therefore, the parameter-free dependency probing method proposed in Wu et al. (2020) might be more preferred.

3 Method

In this section, we first introduce how to induce trees from PTMs, then we describe three tree-based ALSC models, which are selected from three representative methods of incorporating the dependency tree in ALSC task.

3.1 Inducing Tree Structure from PTMs

Perturbed Masking (Wu et al., 2020) can induce trees from the pre-trained models without additional parameters. Generally, a broad range of PTMs can be applied in the Perturbed Masking method. For the sake of being representative and practical, we select BERT and RoBERTa as our base models.

In this subsection, we first briefly introduce the model structure of BERT and RoBERTa, then present the basic idea of the Perturbed Masking method. More details about them can be found in their respective reference papers.

3.1.1 BERT and RoBERTa

BERT (Devlin et al., 2019) and RoBERTa (Liu et al., 2019) both take Transformers (Vaswani et al., 2017) as backbone architecture. Generally, they can be formulated as the following equations

$$\hat{h^l} = \mathrm{LN}(h^{l-1} + \mathrm{MHAtt}(h^{l-1})), \tag{1}$$

$$h^l = \mathrm{LN}(\hat{h^l} + \mathrm{FFN}(\hat{h^l})), \tag{2}$$

where h^0 is the BERT/RoBERTa input representation, formed by the sum of token embeddings, position embeddings, and segment embeddings; LN is the layer normalization layer; MHAtt is the multi-head self-attention; FFN contains three layers, the first one is a linear projection layer, then an activation layer, then another linear projection layer; l is the depth of Transformer layers. The base and large version of BERT and RoBERTa have 12, 24 Transformer layers, respectively.

BERT is pre-trained on Masked Language Modeling (MLM) and Next Sentence Prediction (NSP) tasks. In the MLM task, 15% of the tokens in a sentence are manipulated in three ways. Specifically, 10%, 10%, 80% of them are replaced by a random token, itself, or a "[MASK]" token, respectively. In the NSP task, two sentences A and B are concatenated before sending to BERT. Given 50% of the time when B is the next utterance of A, BERT needs to utilize the vector representation of "[CLS]" to figure out whether the input is continuous or not. RoBERTa is only pre-trained on the MLM task.

3.1.2 Perturbed Masking

Perturbed Masking aims to detect syntactic information from pre-trained models. For a sentence $\mathbf{x} = [x_1, \ldots, x_T]$, BERT and RoBERTa will map each x_i into a contextualized representation $H_\theta(\mathbf{x})_i$. Perturbed Masking is trying to derive the value $f(x_i, x_j)$ that denotes the impact a token x_j has on another token x_i. To derive this value, it first uses the "[MASK]" (or "<mask>" in RoBERTa) to replace the token x_i, which returns a representation $H_\theta(\mathbf{x}\backslash\{x_i\})_i$ for the masked x_i; secondly, it further masks the token x_j, which returns a representation $H_\theta(\mathbf{x}\backslash\{x_i, x_j\})_i$ with both x_i, x_j being masked. The impact value $f(x_i, x_j)$ is calculated by the Euclidean distance as follows,

$$f(x_i,x_j) = ||H_\theta(\mathbf{x}\backslash\{x_i\})_i - H_\theta(\mathbf{x}\backslash\{x_i,x_j\})_i||_2 \tag{3}$$

By repeating this process between every two tokens in the sentence, we can get an impact matrix $\mathbf{M} \in \mathbb{R}^{T\times T}$ and $\mathbf{M}_{i,j} = f(x_i, x_j)$. The tree decoding algorithm, such as Eisner (Eisner, 1996) and Chu–Liu/Edmonds' algorithm (Chu and Liu, 1965; Edmonds, 1967), is then used to extract the dependency tree from the matrix M. The Perturbed Masking can exert on any layer of BERT or RoBERTa.

3.2 ALSC Models Based on Trees

In this subsection, we introduce three representative tree-based ALSC models. Each of the model is from the methods mentioned in the Introduction part (Section 1). For a fair comparison, all the selected models are of the most recently advanced tree-based ALSC models. We briefly introduce these three models as follows.

3.2.1 Aspect-specific Graph Convolutional Networks (ASGCN)

The Aspect-specific Graph Convolutional Networks (ASGCN) is proposed by Sun et al. (2019b). They utilize the dependency tree as a graph, where each word is viewed as a node and the dependencies between words are deemed as edges. After converting the dependency tree into the graph, ASGCN uses the Graph Convolutional Network (GCN) to operate on this graph to model dependencies between each word.

3.2.2 Proximity-Weighted Convolution Network (PWCN)

The Proximity-Weighted Convolution Network (PWCN) model is proposed by Zhang et al. (2019b). They try to help the aspect to find their contextual words. For an input sentence, the PWCN first gets its dependency tree, and based on this tree it would assign a proximity value to each word in the sentence. The proximity value for each word is calculated by the shortest path in the dependency tree between this word and the aspects.

3.3 Relational Graph Attention Network (RGAT)

The Relational Graph Attention Network (RGAT) is proposed by Wang et al. (2020). In the RGAT model, they transform the dependency tree into an aspect-oriented dependency tree. The aspect-oriented dependency tree uses the aspect as the root node, and all other words depend on the aspect directly. The relation between the aspect and other words is either based on the syntactic tag or the tree-based distance in the dependency tree. Specifically, the RGAT reserves syntactic tags for words with 1 tree-based distance to aspect, and assigns virtual tags to longer distance words, such as “2:con” for “A 2 tree-based distance connection”. Therefore, the RGAT model not only exploits the topological structure of the dependency tree but also the tree-based distance between two words.

Dataset	Split	Positive	Negative	Neutral
Rest14	Train	2164	807	637
	Test	728	196	196
Laptop14	Train	994	870	464
	Test	341	128	169
Twitter	Train	1561	1560	3127
	Test	173	173	346

Table 1: Data statistics.

4 Experimental Setup

In this section, we present details about the datasets, the tree structures used in experiments, as well as the experiments implementations. We conduct experiments on all six datasets across four languages. But due to the limited space, we present our experiments on the non-English datasets in the Appendix.

4.1 Datasets

We run experiments on six benchmark datasets. Three of them, namely, Rest14, Laptop14, and Twitter, are English datasets. Rest14 and Laptop14 are from SemEval 2014 task 4 (Pontiki et al., 2014), containing sentiment reviews from restaurant and laptop domains. Twitter is from Dong et al. (2014), which is processed from tweets. The statistics of these datasets are presented in Table 6. Details of the other three non-English datasets can be found in the Appendix. Following previous works, we remove samples with conflicting polarities or with “NULL” aspects in all datasets.

4.2 Tree Structures

For each dataset, we obtain five kinds of trees from three sources. **(1)** The first one is derived from the off-the-shelf dependency tree parser, such as spaCy[2] and allenNLP[3], written as “Dep.”. For the three English datasets, we use the biaffine parser from the allenNLP package to get the dependency tree, which is reported in Wang et al. (2020) that the biaffine parser could achieve better performance. **(2)** We induce trees from the pre-trained BERT and RoBERTa by the Perturbed Masking method (Wu et al., 2020), written them as “BERT Induced Tree” and “RoBERTa Induced Tree”, respectively. **(3)** We

[2] http://spacy.io/

[3] http://www.allennlp.org/

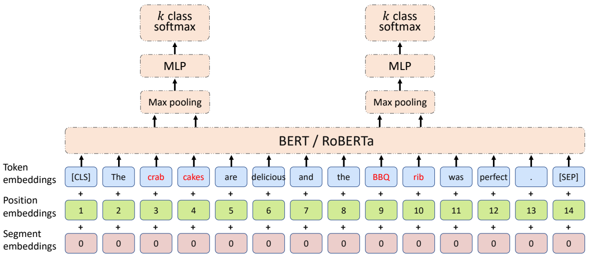


Figure 1: Overall architecture of our fine-tuning model. This structure is enough to achieve SOTA or near SOTA performance in six ALSC datasets based on RoBERTa.

use the Perturbed Masking method to induce trees from the fine-tuned BERT and RoBERTa after fine-tuning in the corresponding datasets. These two are written as "FT-BERT Induced Tree" and "FT-RoBERTa Induced Tree".

Besides, we add "Left-chain" and "Right-chain" in our experiments. "Left-chain", "Right-chain" mean that every word deems its previous or next word as the dependent child word.

4.3 Implementation Details

In order to derive the FT-PTMs Induced Tree, we fine-tune BERT and RoBERTa on the ALSC datasets. To introduce as few parameters as possible, a rather simple MLP is used and the overall structure of our fine-tuning model is presented in Figure 1. The fine-tuning experiments are with the batch size $b = 32$, dropout rate $d = 0.1$, learning rate $\mu = $ 2e-4 using the AdamW optimizer with the default settings.

As for the Perturbed Masking method, we apply Chu–Liu/Edmonds' algorithm for the tree decoding. For the induced trees, we first induce trees from each layer of the PTMs, then test them by the model in Figure 1 on dev set which is composed by 20% of training set. Experiments show that the trees induced from the 11th layer of the PTMs could achieve the best performance among all layers, which is applied for all our experiments.

We conduct multiple experiments incorporating different trees (Section 4.2) into the aforementioned tree-based models (Section 3.2). Specifically, we use the 300-dimension Glove (Pennington et al., 2014) embeddings for English datasets. We keep the word embeddings fixed to avoid overfitting. It is worth noting that in experiments with the RGAT model, since the induced tree does not provide syntactic tags, we assign virtual tags for every dependency in a uniform way, which slightly damage the performance of model.

5 Experimental Results

5.1 ALSC Performance with Different Trees

The comparison between models with different trees is presented in Table 2, which comprises experiments results of English datasets. The results of non-English datasets can be found in the Appendix.

We observe that among all the trees, incorporating FT-RoBERTa Induced Tree leads to the best results on all datasets. On average, models based on the FT-RoBERTa Induced Tree outperform "Dep." by about 1.1% in accuracy. This proves the effectiveness and advantage of FT-RoBERTa Induced Tree in this competitive comparison.

Models using BERT Induced Tree and RoBERTa Induced Tree from Table 2 show small performance difference in all but one dataset, and both are close to the "Left-chain" and "Right-chain" baselines. To have a better sense, we visualize trees induced from RoBERTa in Figure 2b. It shows that RoBERTa Induced Tree has strong neighboring connection dependency pattern. This behavior is expected since the masked language modeling pre-training task will make words favor depending more on its neighboring words. This tendency may be the reason why PTMs induced trees perform similarly to the "Left-chain" and "Right-chain" baselines.

To answer the question **Q1** in the Introduction part (Section 1), we need to compare the "Dep.", BERT Induced Tree, and RoBERTa Induced Tree results. The results show that models with dependency trees usually achieve better performance than PTMs induced trees. This is predictable since the word in PTMs induced trees tends to depend on words in their either left or right side as shown in Figure 2. It is worth noting that this observation does not align with the observation in Wu et al. (2020). The experiments based on PWCN in Wu et al. (2020) show that BERT Induced Tree achieves comparable results with the "Dep.", which is consistent with our PWCN results. However, this observation does not hold when the induced trees are used in a broader range of tree-based ALSC models, especially for the RGAT model in the bottom of Table 2. More detailed analysis will be provided in the next section.

Although models with the PTMs induced trees usually perform worse than those with the dependency parsing trees, models with trees induced from ALSC fine-tuned RoBERTa can surpass both of them. Take RoBERTa Induced Tree and FT-RoBERTa Induced Tree in Table 2 for example,

Model	Tree Features	Tree Structure	Rest14		Laptop14		Twitter	
			Acc.	F_1	*Acc.*	F_1	*Acc.*	F_1
BiLSTM	-	-	77.59	67.05	70.06	64.46	71.39	69.45
ASGCN	Topological Structure	Zhang et al. (2019a)	80.86	72.19	75.55	71.05	72.15	70.40
		Dep.	81.42	72.87	75.54	71.66	72.36	70.32
		Left-chain	80.89	71.92	73.98	69.81	71.96	70.47
		Right-chain[4]	80.89	71.92	73.98	69.81	71.96	70.47
		BERT Induced Tree	81.07	72.87	74.29	70.42	72.39	70.25
		RoBERTa Induced Tree	81.16	72.33	74.76	70.0	72.76	71.17
		FT-BERT Induced Tree	81.87	72.89	74.85	70.71	73.36	71.61
		FT-RoBERTa Induced Tree	**82.31**	**73.53**	**76.33**	**72.76**	**73.84**	**72.66**
PWCN	Tree-based Distance	Zhang et al. (2019b)	80.96	72.21	76.12	72.12	-	-
		Dep.	80.89	72.16	75.86	71.94	72.10	70.75
		Left-chain	80.78	72.37	73.35	69.41	71.24	69.42
		Right-chain[4]	80.78	72.37	73.35	69.41	71.24	69.42
		BERT Induced Tree	80.98	72.04	73.82	69.35	72.10	69.90
		RoBERTa Induced Tree	81.16	73.20	73.98	69.94	72.11	70.74
		FT-BERT Induced Tree	81.33	73.57	74.96	70.93	72.54	70.75
		FT-RoBERTa Induced Tree	**82.40**	**73.69**	**76.95**	**73.21**	**73.84**	**71.43**
RGAT	Structure & Distance	Wang et al. (2020)	**83.30**	**76.08**	77.42	73.76	**75.57**	73.82
		Dep.	82.14	74.62	76.49	72.63	74.57	72.57
		Left-chain	80.53	69.63	74.14	70.04	73.41	71.99
		Right-chain[4]	80.53	69.63	74.14	70.04	73.41	71.99
		BERT Induced Tree	81.27	71.76	75.23	70.47	73.49	72.19
		RoBERTa Induced Tree	81.42	71.79	75.36	71.11	73.78	72.37
		FT-BERT Induced Tree	81.60	72.48	75.96	71.96	74.13	72.47
		FT-RoBERTa Induced Tree	82.76	75.25	**77.43**	**74.21**	75.43	**74.04**

Table 2: The performance(%) of tree-based ALSC models incorporating different tree structures on three major English datasets. Following previous work, Accuracy(*Acc.*) and Marco-F_1(F_1) are used for metric. The reported results are averaged by 3 runs with random initialization. Results named as cited format refer to performance reported in the original paper. Dep. refers to the dependency tree generated from the well-known Biaffine Parser (Dozat and Manning, 2017). As mentioned in Section 4.2, BERT Induced Tree, RoBERTa Induced Tree, FT-BERT, and FT-RoBERTa Induced Tree refer to tree structures induced from corresponding PTM. We provide BiLSTM since the other three are different tree-based models over BiLSTM. We highlight the best results of each model in bold.

compared with RoBERTa Induced Tree, models incorporating FT-RoBERTa Induced Tree achieves an average accuracy improvement of 1.56%. This trending is also observed between BERT Induced Tree and FT-BERT Induced Tree.

Tree Structure	Rest14	Laptop14	Twitter
Dep.	0.509	0.500	0.509
Left-chain	1.000	1.000	1.000
Right-chain	1.000	1.000	1.000
BERT	0.710	0.690	0.741
RoBERTa	0.702	0.705	0.722
FT-BERT	0.606	0.519	0.666
FT-RoBERTa	**0.506**	**0.480**	**0.485**

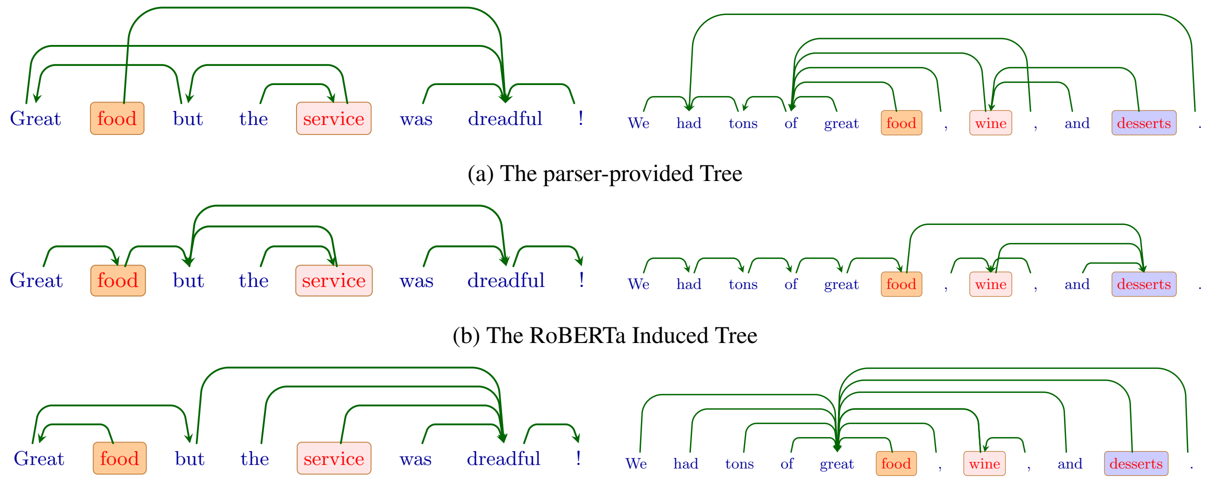
Table 3: Proportion of neighboring connections of different trees in all datasets. We use the short name of induced trees here as well as Table 4 and Table 5.

5.2 Analysis

To further investigate the reasons for the difference between trees, we propose a set of quantitative metrics, presented in Table 3 and Table 4.

The **Proportion of Neighboring Connections** is to calculate the proportion of neighboring connections in the sentence, shown in Table 3. A neighboring connection links the word to its left/right neighbor word. From Table 3, we observe that on average over 70% relations in BERT/RoBERTa Induced Tree are neighboring connections. This will damage the performance of models using topological structures of trees. Thus, PTMs induced trees usually perform worse than "Dep.", with a slight

[4] The Left/Right-chain are exactly the same input files after the data preprocessing in these three models.


Great food but the service was dreadful !
We had tons of great food , wine , and desserts .
Great food but the service was dreadful !
We had tons of great food , wine , and desserts .
Great food but the service was dreadful !
We had tons of great food , wine , and desserts .

(a) The parser-provided Tree

(b) The RoBERTa Induced Tree

(c) The FT-RoBERTa Induced Tree

Figure 2: Visualization of different trees. The colored box refers to the aspect terms. Since ROOT has no directional relation arcs, we omit the ROOT notation here. For the same two sentences, trees from dependency parser, RoBERTa and fine-tuned RoBERTa are displayed. As Figure 2b shows, trees induced from RoBERTa tend to have more neighboring connections. As the bottom two figures show, trees induced from fine-tuned RoBERTa tend to have connections between sentiment words and others words.

improvement over left/right-chains.

In comparison with RoBERTa Induced Tree, a significant decline of the proportion is shown in FT-RoBERTa Induced Tree in Table 3. We see the same tendency in BERT Induced Tree and FT-BERT Induced Tree. This marks the consistent structure change in the fine-tuning process, indicating the transition to a more diverse structure. As shown in Figure 2b, RoBERTa Induced Tree has a clear pattern to depend on words in their neighbor side. Yet FT-RoBERTa Induced Tree in Figure 2c shows a more diverse dependency pattern.

Aspects-sentiment Distance is the average distance between aspect and sentiment words. We pre-define a sentiment words set C. For a sentence S_i in datasets S, the set of aspects words in S_i is termed as w. $S_i \cap C$ is the set of sentiment words appearing both in the sentence S_i and the sentiment words set C. The Aspects-sentiment Distance(AsD) is calculated as follows:

$$AsD(S_i) = \frac{\sum\limits_{w}^{w_i} \sum\limits_{C'=S_i \cap C}^{C'_i} dist(C'_i, w_i)}{|w|\,|C'|} \tag{4}$$

$$AsD = \frac{\sum\limits_{S}^{S_i} AsD(S_i)}{|S|} \tag{5}$$

where $|\cdot|$ is the number of elements in the set and $dist(x_i, x_j)$ represents the relative distance between x_i and x_j in the tree. Specifically, C contains sentiment words counted on Amazon-2 from Tian et al. (2020), which can be found in the Appendix. As for the Rest14 and Laptop14, Xu et al. (2020) provides the paired sentiment words with its corresponding aspect. We also calculate the paired Aspects-sentiment Distance(pAsD) on these two datasets, which only counts the distance between aspect and its corresponding sentiment words.

Tree Structure	Rest14	Laptop14	Twitter
Dep.	4.46 / 3.19	3.77 / 3.13	4.26
Left-chain	7.49 / 6.06	6.48 / 5.97	7.90
Right-chain	7.49 / 6.06	6.48 / 5.97	7.90
BERT	5.85 / 4.20	5.06 / 4.19	5.87
RoBERTa	5.05 / 3.61	4.49 / 3.67	5.39
FT-BERT	3.85 / 3.58	3.65 / 3.22	5.06
FT-RoBERTa	**3.56 / 2.92**	**3.35 / 2.88**	**3.55**

Table 4: The Aspects-sentiment Distance of different trees in all datasets. The less result indicates shorter distance between aspects and sentiment words. The values of Rest14 and Laptop14 are formed like “pAsD / AsD”.

We present the Aspects-sentiment Distance (AsD) of different trees in English datasets in Table 4. Results show that FT-RoBERTa has the least AsD value, indicating the shortest aspects-sentiment distance. Compared to PTMs induced trees, the trees from FT-PTMs have less AsD, indicating shortened aspects-sentiment distance. This shows that the FT-PTMs induced

Embedding	Model	Tree Structure	Rest14		Laptop14		Twitter	
			Acc.	F_1	*Acc.*	F_1	*Acc.*	F_1
Static Embedding	**BiLSTM** †	-	77.59	67.05	70.06	64.46	71.39	69.45
	LSTM+SynATT $^{\sharp}$	Dep.	80.45	71.26	72.57	69.13	-	-
	AdaRNN $^{\sharp}$	Dep.	-	-	-	-	66.30	65.90
	TD-GAT $^{\sharp}$	Dep.	80.35	76.13	74.13	72.01	72.68	71.15
BERT	**MLP**	-	85.35	78.38	78.36	74.16	75.92	74.41
	DGEDT $^{\sharp}$	Dep.	86.30	80.0	79.80	75.60	**77.90**	75.40
	RGAT $^{\sharp}$	Dep.	86.60	81.35	78.21	74.07	76.15	74.88
	RACL$^{\sharp}$	-	-	**81.61**	-	73.91	-	81.61
RoBERTa	**MLP**	-	87.37	80.96	83.78	80.73	77.17	**76.20**
	RoBERTa-ASC $^{\sharp}$	Dep.	82.82	75.12	74.12	70.52	-	-
	LCFS-ASC-CDW $^{\sharp}$	Dep.	86.71	80.31	80.52	77.13	-	-
	ASGCN	Dep.	86.90	80.75	81.66	78.31	75.28	74.38
		FT-RoBERTa	86.87	80.59	83.33	80.32	76.10	75.07
	PWCN	Dep.	87.41	81.07	**84.16**	**81.18**	76.63	75.60
		FT-RoBERTa	87.35	80.85	84.01	81.08	77.02	75.52
	RGAT	Dep.	87.43	80.61	83.43	80.28	74.42	72.93
		FT-RoBERTa	**87.52**	81.29	83.33	79.95	75.81	74.91

Table 5: The results(%) of SOTA ALSC models on English datasets. The results with “†” are retrieved from Sun et al. (2019b), and those with “♯” are retrieved from the original papers. Those without additional symbols are on our own. We highlight the best results on bold.

trees are more sentiment-word-oriented, which partially reveals that the fine-tuning in ALSC encourages the aspects to find sentiment words. However, for the “Dep.”, we notice that some Twitter results in Table 2 can not be fully explained by these two proposed metrics. We conjecture that the grammar casualness features the Twitter corpus, which makes the parser hard to provide an accurate dependency parsing tree. Still, these two metrics can be suitable for the induced trees.

Taken together, as the conclusion to **Q2**, these analyses demonstrate that the fine-tuning on ALSC could adapt the induced tree implicitly. On the one hand, less proportion of neighboring connections after fine-tuning indicates the increase of long range connections. On the other hand, less Aspects-sentiment Distance after fine-tuning illustrates the shorter distance between aspects and sentiment words, which helps to model connections between aspects and sentiment words. Thus, as shown in Section 5.1, fine-tuning RoBERTa in ALSC not only makes induced tree better suit the ALSC task but also outperform the dependency tree when combined with different tree-based ALSC models.

5.3 Comparison between ALSC models

Additional, we explore how well the fine-tuned RoBERTa model could achieve in the ALSC task. We select a set of top high-performing models of ALSC as state-of-the-art alternatives. The comparison results are shown in Table 5.

Comparing with all these SOTA alternatives, surprisingly, the RoBERTa with an MLP layer achieve SOTA or near SOTA performance. Especially, compared to other datasets, we notice that significant improvement is obtained on the Laptop14 dataset. We assume that the pre-training corpus of RoBERTa may be more friendly to the laptop domain since the RoBERTa-MLP already obtains much better results than the BERT-MLP on Laptop14. For these BERT-based models in the second row of Table 5, similar experiments using RoBERTa are conducted. However, limited improvements have been made over the RoBERTa-MLP. We expect that induced trees from models specifically pre-trained for ALSC (Tian et al., 2020) may provide more information, which is left for the future works.

The FT-RoBERTa Induced Tree could be beneficial to Glove based ALSC models. However, incorporating trees over the RoBERTa brings no significant improvement, even the decline can be seen in some cases. This may be caused by failure to reconcile the implicitly entailed tree with external tree. We argue that incorporating trees over the RoBERTa in currently widely-used tree methods may be the loss outweighs the gain. Additionally, in the review of previous ALSC works, we notice that very few works employ the RoBERTa as the base model. We would attribute this to the difficulty of optimizing the RoBERTa-based ALSC models. As the higher architecture, which is usually randomly initialized, needs a bigger learning rate compared to the RoBERTa. The inappropriate hyperparameters may be the cause reason for the lagging performance of previous RoBERTa-based ALSC works (Phan and Ogunbona, 2020).

6 Conclusion

In this paper, we analyze several tree structures for the ALSC task including parser-provided dependency tree and PTMs-induced tree. Specifically, we induce trees using the Perturbed Masking method from the original PTMs and ALSC fine-tuned PTMs respectively, and then compare the different tree structures on three typical tree-based ALSC models on six datasets across four languages. Experiments reveal that fine-tuning on ALSC task forces PTMs to implicitly learn more sentiment-word-oriented trees, which can bring benefits to Glove based ALSC models. Benefited from its better implicit syntactic information, the fine-tuned RoBERTa with an MLP is enough to obtain SOTA or near SOTA results for ALSC task. Our work can lead to several promising directions, such as PTMs-suitable tree-based models and better tree-inducing methods from PTMs.

Acknowledgment

We would like to thank the anonymous reviewers for their helpful comments. This work was supported by the National Key Research and Development Program of China (No. 2020AAA0106700) and National Natural Science Foundation of China (No. 62022027).

References

Danqi Chen and Christopher Manning. 2014. A fast and accurate dependency parser using neural networks. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 740–750, Doha, Qatar. Association for Computational Linguistics.

Muhao Chen, Yingtao Tian, Haochen Chen, Kai-Wei Chang, Steven Skiena, and Carlo Zaniolo. 2019. Learning to represent bilingual dictionaries. In *Proceedings of the 23rd Conference on Computational Natural Language Learning (CoNLL)*, pages 152–162, Hong Kong, China. Association for Computational Linguistics.

Peng Chen, Zhongqian Sun, Lidong Bing, and Wei Yang. 2017. Recurrent attention network on memory for aspect sentiment analysis. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 452–461, Copenhagen, Denmark. Association for Computational Linguistics.

Y. J. Chu and T. H. Liu. 1965. On the shortest arborescence of a directed graph. *Science Sinica*, 14.

Kevin Clark, Urvashi Khandelwal, Omer Levy, and Christopher D. Manning. 2019. What does BERT look at? an analysis of BERT's attention. In *Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 276–286, Florence, Italy. Association for Computational Linguistics.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451, 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 *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 49–54, Baltimore, Maryland. Association for Computational Linguistics.

Timothy Dozat and Christopher D. Manning. 2017. Deep biaffine attention for neural dependency parsing. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net.

Jack Edmonds. 1967. Optimum branchings. *Journal of Research of the national Bureau of Standards B*, 71(4):233–240.

Jason M. Eisner. 1996. Three new probabilistic models for dependency parsing: An exploration. In *COLING 1996 Volume 1: The 16th International Conference on Computational Linguistics*.

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, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

John Hewitt and Percy Liang. 2019. Designing and interpreting probes with control tasks. 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 2733–2743, Hong Kong, China. Association for Computational Linguistics.

John Hewitt and Christopher D. Manning. 2019. A structural probe for finding syntax in word representations. 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 4129–4138, Minneapolis, Minnesota. Association for Computational Linguistics.

Binxuan Huang and Kathleen 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 5469–5477, Hong Kong, China. Association for 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.

Svetlana Kiritchenko, Xiaodan Zhu, Colin Cherry, and Saif Mohammad. 2014. NRC-Canada-2014: Detecting aspects and sentiment in customer reviews. In *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*, pages 437–442, Dublin, Ireland. Association for Computational Linguistics.

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

Jiangming Liu and Yue Zhang. 2017. Attention modeling for targeted sentiment. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 572–577, Valencia, Spain. Association for Computational Linguistics.

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

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

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. GloVe: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar. Association for Computational Linguistics.

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, Online. Association for Computational Linguistics.

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

Maria Pontiki, Dimitris Galanis, 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.

Xipeng Qiu, TianXiang Sun, Yige Xu, Yunfan Shao, Ning Dai, and Xuanjing Huang. 2020. Pre-trained models for natural language processing: A survey. *SCIENCE CHINA Technological Sciences*.

Sebastian Ruder, Parsa Ghaffari, and John G. Breslin. 2016. A hierarchical model of reviews for aspect-based sentiment analysis. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 999–1005, Austin, Texas. Association for Computational Linguistics.

Chi Sun, Luyao Huang, and Xipeng Qiu. 2019a. Utilizing bert for aspect-based sentiment analysis via constructing auxiliary sentence. 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 380–385.

Kai Sun, Richong Zhang, Samuel Mensah, Yongyi Mao, and Xudong Liu. 2019b. 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 5679–5688, Hong Kong, China. Association for Computational Linguistics.

Duyu Tang, Bing Qin, Xiaocheng Feng, and Ting Liu. 2016a. Effective LSTMs for target-dependent sentiment classification. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 3298–3307, Osaka, Japan. The COLING 2016 Organizing Committee.

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

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

Jialong Tang, Ziyao Lu, Jinsong Su, Yubin Ge, Linfeng Song, Le Sun, and Jiebo Luo. 2019. Progressive self-supervised attention learning for aspect-level sentiment analysis. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 557–566, Florence, Italy. 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 Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018*, pages 5956–5963. AAAI Press.

Hao Tian, Can Gao, Xinyan Xiao, Hao Liu, Bolei He, Hua Wu, Haifeng Wang, and Feng Wu. 2020. SKEP: Sentiment knowledge enhanced pre-training for sentiment analysis. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4067–4076, Online. Association for Computational Linguistics.

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

Duy-Tin Vo and Yue Zhang. 2015. Target-dependent twitter sentiment classification with rich automatic features. In *Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence, IJCAI 2015, Buenos Aires, Argentina, July 25-31, 2015*, pages 1347–1353. AAAI Press.

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, Online. Association for Computational Linguistics.

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

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.

Zhiyong Wu, Yun Chen, Ben Kao, and Qun Liu. 2020. Perturbed masking: Parameter-free probing for analyzing and interpreting BERT. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4166–4176, Online. Association for Computational Linguistics.

Lu Xu, Hao Li, Wei Lu, and Lidong Bing. 2020. Position-aware tagging for aspect sentiment triplet extraction. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2339–2349, Online. Association for Computational Linguistics.

Wei Xue and Tao Li. 2018. Aspect based sentiment analysis with gated convolutional networks. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2514–2523, Melbourne, Australia. Association for Computational Linguistics.

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 4568–4578, Hong Kong, China. Association for Computational Linguistics.

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, SIGIR 2019, Paris, France, July 21-25, 2019*, pages 1145–1148. ACM.

Meishan Zhang, Yue Zhang, and Duy-Tin Vo. 2016. Gated neural networks for targeted sentiment analysis. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, February 12-17, 2016, Phoenix, Arizona, USA*, pages 3087–3093. AAAI Press.

Yaowei Zheng, Richong Zhang, Samuel Mensah, and Yongyi Mao. 2020. Replicate, walk, and stop on syntax: An effective neural network model for aspect-level sentiment classification. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*, pages 9685–9692. AAAI Press.

A Experiments on non-English Datasets

In this section, we provide details about our experiments on non-English datasets.

A.1 Datasets

We conduct experiments on three non-English datasets, which are named Dutch, French, and Spanish, respectively. All of them are restaurant review datasets from SemEval-2016 task 5 (Pontiki et al., 2016), whose languages are the same as dataset names. Detailed data statistics can be found in Table 6. Following previous works, we remove samples with conflicting polarities or with "NULL" aspect terms in all datasets.

Dataset	Split	Positive	Negative	Neutral
Dutch	Train	720	386	108
	Test	229	120	23
French	Train	833	683	98
	Test	320	253	54
Spanish	Train	1308	443	79
	Test	505	171	33

Table 6: Data statistics.

A.2 Tree Structures

We obtain five kinds of trees for every dataset. The first one is to use the off-the-shelf dependency tree parser to get parser-provided dependency trees, written as "Dep.". Specifically, we utilize the spaCy parser for the non-English datasets. The second method is to induce the trees from the pre-trained mBERT and XLM-R (Conneau et al., 2020) base models by the Perturbed Masking method (Wu et al., 2020), written them as "BERT Induced Tree" and "RoBERTa Induced Tree", respectively. The third method is to use the same method as above to induce trees from the mBERT and XLM-R after fine-tuning in the corresponding datasets with the same model structure as English datasets. These two are written as "FT-BERT Induced Tree" and "FT-RoBERTa Induced Tree" to have a uniform form as the English datasets. Similarly, we add "Left-chain" and "Right-chain" as baselines. "Left-chain", "Right-chain" mean that every word deems its previous or next word as the dependent child word.

A.3 Implementation Details

Similar to the English datasets, Experiments incorporating tree-based ALSC models with different trees are conducted on non-English datasets, as well as the fine-tuning of PLMs. All experiments are conducted on the NVIDIA GTX1080Ti.

For experiments with tree-based models, we use the 300-dimension pre-trained embeddings (Ruder et al., 2016) for non-English datasets. We keep the word embeddings fixed to avoid overfitting. Other parameters are initialized with original models. It is worth noting that in RGAT Model reproduction, since the induced tree does not provide relation labels, we assign virtual relations for every dependency in a uniform way.

We retain the fine-tuning experiments with batch size $b = 32$, dropout rate $d = 0.1$, learning rate $\mu = $ 2e-4 using the AdamW optimizer with the default settings.

As for the induced trees, We choose the trees induced from the 11th layer in all of our experiments.

A.4 Experimental Results

A.4.1 ALSC Performance with Different Trees

The comparison between models with different trees is presented in Table 7, which comprises experiments results of non-English datasets. Experimental results shows that: (1) Incorporating FT-RoBERTa Induced Tree leads to the best results on all datasets, which proves the effectiveness and advantage of FT-RoBERTa Induced Tree in non-

Model	Tree Features	Tree Sturcture	Dutch		French		Spanish	
			Acc.	F_1	*Acc.*	F_1	*Acc.*	F_1
BiLSTM	-	-	83.30	62.50	80.0	67.50	85.30	62.10
ASGCN	Topological Structure	Dep.	84.18	70.06	79.23	65.0	87.6	67.36
		BERT Induced Tree	84.45	67.25	79.23	66.31	87.10	67.58
		FT-BERT Induced Tree	83.37	68.12	79.38	62.27	86.70	69.07
		RoBERTa Induced Tree	84.45	**70.94**	79.53	67.20	86.70	68.19
		FT-RoBERTa Induced Tree	**84.99**	68.26	**80.31**	**67.4**	**87.8**	**72.88**
PWCN	Relative Distance	Dep.	83.38	67.82	79.23	66.28	86.25	**67.95**
		BERT Induced Tree	84.18	67.37	78.46	64.6	87.09	66.57
		FT-BERT Induced Tree	84.18	68.17	78.62	66.57	86.53	67.87
		RoBERTa Induced Tree	84.90	68.30	78.62	63.27	85.97	66.38
		FT-RoBERTa Induced Tree	**85.25**	**70.21**	**80.0**	**67.9**	**87.23**	64.93
RAGT	Structure & Distance	Dep.	84.45	59.85	79.53	66.16	86.14	56.44
		BERT Induced Tree	84.45	57.36	76.92	58.14	86.53	61.70
		FT-BERT Induced Tree	84.18	59.67	78.61	60.79	85.50	62.66
		RoBERTa Induced Tree	84.71	67.60	78.15	61.10	86.81	61.88
		FT-RoBERTa Induced Tree	**85.25**	**69.53**	**81.38**	**66.97**	**87.37**	**65.30**

Table 7: The averaged performance(%) of tree-based ALSC models incorporating different tree structures on three non-English datasets. Dep. refers to the dependency tree generated by spaCy. As mentioned in English datasets, BERT Induced Tree, RoBERTa Induced Tree, FT-BERT, and FT-RoBERTa Induced Tree refer to tree structures extracted from corresponding PLMs.

Embedding	Model	Tree Structure	Dutch		French		Spanish	
			Acc.	F_1	*Acc.*	F_1	*Acc.*	F_1
Static Embedding	**BiLSTM**	-	83.3	62.5	80.0	67.5	85.3	62.1
	SA-LSTM-P ♯	-	87.3	-	82.4	-	88.0	-
	Our ASGCN	Dep.	81.6	61.0	75.5	63.0	85.0	59.0
	Our RGAT	Dep.	81.0	62.1	75.1	53.3	84.6	55.2
	Our PWCN	Dep.	84.1	69.2	78.4	66.7	86.9	67.5
mBERT	**MLP**	-	80.37	63.43	78.06	65.04	88.21	68.03
XLM-R	**MLP**	-	**88.36**	**76.29**	85.95	74.72	91.48	77.96
	ASGCN	Dep.	87.97	74.38	86.43	77.14	91.91	77.49
		FT-RoBERTa	88.2	75.23	86.04	76.21	**92.47**	**78.74**
	PWCN	Dep.	88.36	75.72	86.4	76.8	91.51	77.32
		FT-RoBERTa	88.1	75.54	**86.69**	**77.42**	91.44	78.13
	RGAT	Dep.	88.31	70.57	85.92	75.14	91.61	76.41
		FT-RoBERTa	87.86	70.97	86.41	74.38	92.11	76.62

Table 8: The results(%) of ALSC models incorporating with different tree structures on non-English datasets. The definition of tree structures retains the same as the aforementioned. The results with "♯" are retrieved from the original papers.

English datasets. Moreover, we find that the results of the FT-RoBERTa Induced Tree usually have more stable F_1 scores. (2) Subjected to the quality of the parser of non-English languages, models using the PLMs induced trees achieve slightly better performance compared to “Dep.”. This illustrates that the dependency tree could be very sensitive to parser and quality of corpus. (3) Similarly, from “RoBERTa Induced Tree” and “FT-RoBERTa Induced Tree”, we conclude that fine-tuning can substantially enhance the ALSC performance through trees induced from PLMs.

A.4.2 Comparison between ALSC models

Similarly, we compare the performance between the fine-tuned XLM-R and a set of top high-performing models. The results are presented in Table 8. We could see that XLM-R with an MLP is enough to achieve SOTA or near SOTA results in non-English datasets.

B Sentiment words set

positive sentiment words	great, good, like, just, will, well, even, love, best, better, back, want, recommend, worth, easy, sound, right, excellent, nice, real, fun, sure, pretty, interesting, stars
negative sentiment word	too, little, bad, game, down, long, hard, waste, disappointed, problem, try, poor, less, boring, worst, trying, wrong, least, although, problems, cheap

Table 9: The sentiment words used in our analysis, derived from Tian et al. (2020).

To calculate the Aspects-sentiment Distance of different tree structures on English datasets, we predefine a set of sentiment words, shown in Table 9. Specifically, we use the sentiment words described in Tian et al. (2020), which are the selected 50 most frequent sentiment words counted on Amazon-2.