Aspect-based Sentiment Analysis with Opinion Tree Generation

Xiaoyi Bao[1], **Wang Zhongqing**[1] *, **Xiaotong Jiang**[1], **Rong Xiao**[2], and **Shoushan Li**[1]
[1] Natural Language Processing Lab, Soochow University, Suzhou, China
[2] Alibaba Group, Hangzhou, China
baoxiaoyiharris@foxmail.com, devjiang@outlook.com
xiaorong.xr@taobao.com, {wangzq,lishoushan}@suda.edu.cn

Abstract

Existing studies usually extract the sentiment elements by decomposing the complex structure prediction task into multiple subtasks. Despite their effectiveness, these methods ignore the semantic structure in ABSA problems and require extensive task-specific designs. In this study, we introduce a new Opinion Tree Generation model, which aims to jointly detect all sentiment elements in a tree. The opinion tree can reveal a more comprehensive and complete aspect-level sentiment structure. Furthermore, we employ a pre-trained model to integrate both syntax and semantic features for opinion tree generation. On one hand, a pre-trained model with large-scale unlabeled data is important for the tree generation model. On the other hand, the syntax and semantic features are very effective for forming the opinion tree structure. Extensive experiments show the superiority of our proposed method. The results also validate the tree structure is effective to generate sentimental elements.

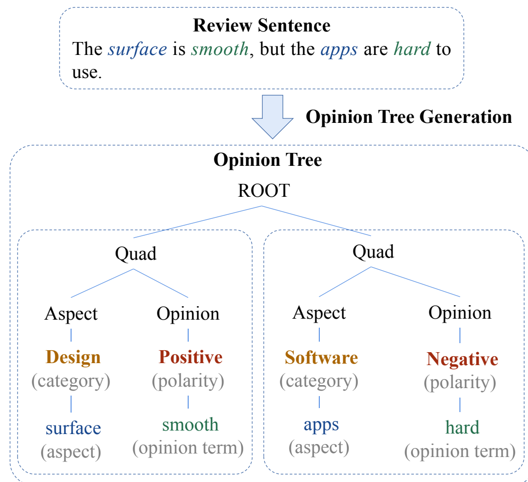


Figure 1: Example of opinion tree generation.

1 Introduction

As a fine-grained sentiment analysis task, aspect-based sentiment analysis (ABSA) has received continuous attention. Multiple fundamental sentiment elements are involved in ABSA, including the aspect term, opinion term, aspect category, and sentiment polarity. Given a simple example sentence "The surface is smooth.", the corresponding elements are "surface", "smooth", "design" and "positive", respectively.

In the literature, the main research line of ABSA focuses on the identification of those sentiment elements such as extracting the aspect term [Qiu *et al.*, 2011], classifying the sentiment polarity for a given aspect [Tang *et al.*, 2016], or jointly predicting multiple elements simultaneously [Peng *et al.*, 2020; Cai *et al.*, 2021]. In general, most ABSA tasks are formulated as either sequence-level or token-level classification problems. However, these methods would suffer severely from error propagation because the overall prediction performance hinges on the accuracy of every step [Peng *et al.*, 2020]. Besides, these methods ignore the label semantics, since they treat the labels as number indices during training [Yan *et al.*, 2021; Zhang *et al.*, 2021b].

Therefore, recent studies tackle the ABSA problem in a unified generative approach. For example, they treat the class index [Yan *et al.*, 2021] or the desired sentiment element sequence [Zhang *et al.*, 2021b] as the target of the generation model. [Zhang *et al.*, 2021a] further rewrite sentiment quads into paraphrase, and employs a paraphrase generation model to generate the sentiment-aware paraphrase. These studies can fully utilize the rich label semantics by encoding the natural language label into the target output.

Despite giving strong empirical results, the generative approaches can suffer from structural guarantees in their neural semantic representation [Lu *et al.*, 2021; Zhou *et al.*, 2021], i.e. they can not capture the semantic structure between aspect terms and opinion words. Intuitively, such issues can be alleviated by having a structural representation of semantic information, which treats aspect terms and opinion words as nodes, and builds structural relations between nodes. Explicit structures are also more interpretable compared to neural representation and have been shown useful for many NLP applications [Xue and Li, 2018; Zhang and Qian, 2020; Wang *et al.*, 2020].

* Corresponding author

In this study, we introduce a new Opinion Tree Generation model, which aims to jointly detect all sentiment elements in a tree for a given review sentence. The opinion tree can be considered as a semantic representation in order to better represent the structure of sentiment elements. As shown in Figure 1, the opinion tree models a sentence using a rooted directed acyclic graph, highlighting its main elements (e.g. aspect terms, opinion words) and semantic relations. It can thus potentially reveal a more comprehensive and complete aspect-level semantic structure for extracting sentiment elements.

Meanwhile, the structure of opinion tree is hard to capture, since it consists of rich semantic relations and multiple interactions between sentiment elements. To this end, we design two strategies for effectively forming the opinion tree structure. Firstly, we propose a constrained decoding algorithm, which can guide the generation process using opinion schemas. In this way, the opinion knowledge can be injected and exploited during inference. Secondly, we explore sequence-to-sequence joint learning of several pre-training tasks to integrate syntax and semantic features and optimize for the performance of opinion tree generation.

Detailed evaluation shows that our model significantly advances the state-of-the-art performance on several benchmark datasets. The results also show that the effect of the proposed opinion tree architecture, and our proposed sequence-to-sequence pre-trained system is necessary to achieve strong performance.

2 Related Work

Aspect-based sentiment analysis (ABSA) has drawn wide attention during the last decade. Early studies focus on the prediction of a single element, such as extracting the aspect term [Qiu *et al.*, 2011], detecting the mentioned aspect category [Bu *et al.*, 2021], and predicting the sentiment polarity for a given aspect [Tang *et al.*, 2016].

Some works further consider the joint detection of two sentiment elements, including the pairwise extraction of aspect and opinion term [Xu *et al.*, 2020b]; the prediction of aspect term and its corresponding sentiment polarity [Zhang and Qian, 2020]; and the co-extraction of aspect category and sentiment polarity [Cai *et al.*, 2020]. Recently, aspect sentiment triplet and quadruple prediction tasks are proposed in ABSA, they employ end-to-end models to predict the sentiment elements in triplet or quadruple format [Peng *et al.*, 2020; Wan *et al.*, 2020; Cai *et al.*, 2021; Zhang *et al.*, 2021a].

More recently, there are some attempts on tackling ABSA problem in a sequence-to-sequence manner [Zhang *et al.*, 2021a], either treating the class index [Yan *et al.*, 2021] or the desired sentiment element sequence [Zhang *et al.*, 2021b] as the target of the generation model. For example, [Yan *et al.*, 2021] treated the ABSA as a text generation problem, and employ a sequence-to-sequence pre-trained model to generate the sequence of aspect terms and opinion words directly. Meanwhile, [Zhang *et al.*, 2021a] proposed a paraphrase model that utilized the knowledge of the pre-trained model via casting the original task to a paraphrase generation process. They employed the paraphrase to represent aspect-based quads.

Different from previous studies, we propose a new task called opinion tree generation, which aims to jointly detect all sentiment elements in a tree for a given review sentence. The opinion tree can reveal a more comprehensive and complete aspect-level sentiment structure for generating sentiment elements.

3 Opinion Tree Generation

Given a review sentence, we aim to predict all aspect-level sentiment quadruplets which correspond to the aspect category, aspect term, opinion term, and sentiment polarity, respectively [Cai *et al.*, 2021; Zhang *et al.*, 2021a]. In this study, we propose an opinion tree generation model to jointly detect all sentiment quadruplets in an opinion tree. As shown in Figure 2, we firstly propose a tree generation model to generate an opinion tree with a constrained decoding algorithm, which can guide the generation process using opinion schemas. We then explore joint learning of several pre-training tasks to integrate syntax and semantic features for forming the opinion tree structure. Afterward, the sentiment quadruplets can be recovered from the opinion tree easily. In the present section, we first introduce how to reformulate ABSA as opinion tree generation via structure linearization, then describe the tree generation model and the constrained decoding algorithm. The joint pre-trained model will be described in the next section.

3.1 Opinion Tree Construction and Linearization

As shown in Figure 2, the opinion tree models a sentence using a rooted directed acyclic graph, highlighting its main elements (e.g. aspect terms, opinion words) and semantic relations. Given a review sentence, we convert aspect sentiment quads (Figure 2c) into opinion tree (Figure 2b) as below:

- We first create a quad node to denote an aspect sentiment quad, and all the quad nodes are connected with a virtual root node;
- The aspect node and opinion node are connected with the corresponding quad node;
- The aspect category is connected with an aspect node, and the sentiment polarity is connected with an opinion node;
- The text spans (i.e, aspect term and opinion word) from the review sentence are linked to the corresponding nodes (i.e., aspect category and sentiment polarity) as leaves.

From Figure 2b, we can see that: the connections between aspect term and aspect category can be used to identify the category of aspect term, and the connection between opinion word and polarity is very helpful for predicting the polarity. In addition, the aspect and opinion nodes are used to separate the aspect term and opinion word. Furthermore, the quad node is used to denote each aspect sentiment quad with corresponding elements.

Since it is much easier to generate a sequence than generate a tree [Xu *et al.*, 2020a; Lu *et al.*, 2021], we linearize the

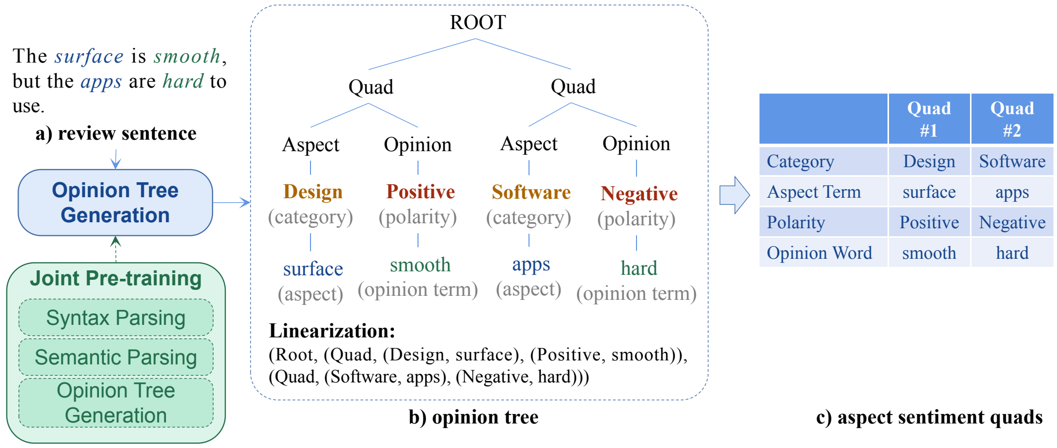


Figure 2: Overview of proposed model.

opinion tree to the target sequence. Given the converted opinion tree, we linearize it into a token sequence (Figure 2b) via depth-first traversal, where "(" and ")" are structure indicators used to represent the structure of linear expressions [Vinyals *et al.*, 2015; van Noord and Bos, 2017]. The traversal order of the same depth is the order in which the text spans appear in the text, e.g., first "surface" then "smooth" in Figure 2b.

3.2 Tree Generation Model

We then employ a tree generation model to generate the linearized opinion tree from the review sentence.

In this study, we employ a sequence-to-sequence model to generate the opinion tree via a transformer-based encoder-decoder architecture [Vaswani *et al.*, 2017]. Given the token sequence $x = x_1, ..., x_{|x|}$ as input, the sequence-to-sequence model outputs the linearized representation $y = y_1, ..., y_{|y|}$. To this end, the sequence-to-sequence model first computes the hidden vector representation $H = h_1, ..., h_{|x|}$ of the input via a multi-layer transformer encoder:

$$H = \mathrm{Encoder}(x_1, ..., x_{|x|}) \tag{1}$$

where each layer of $\mathrm{Encoder}$ is a transformer block with the multi-head attention mechanism.

After the input token sequence is encoded, the decoder predicts the output structure token-by-token with the sequential input tokens' hidden vectors. At the i-th step of generation, the self-attention decoder predicts the i-th token y_i in the linearized form, and decoder state h_i^d as:

$$y_i, h_i^d = \mathrm{Decoder}([H; h_1^d, ..., h_{i-1}^d], y_{i-1}) \tag{2}$$

where each layer of $\mathrm{Decoder}$ is a transformer block that contains self-attention with decoder state h_i^d and cross-attention with encoder state H.

The generated output structured sequence starts from the start token "$\langle bos \rangle$" and ends with the end token "$\langle eos \rangle$". The conditional probability of the whole output sequence $p(y|x)$ is progressively combined by the probability of each step $p(y_i|y_{<i}, x)$:

$$p(y|x) = \prod_{i=1}^{|y|} p(y_i|y_{<i}, x) \tag{3}$$

where $y_{<i} = y_1...y_{i-1}$, and $p(y_i|y_{<i}, x)$ are the probabilities over target vocabulary V normalized by $\mathrm{softmax}$.

Since all tokens in linearized representations are also natural language words, we adopt the pre-trained language model T5 [Raffel *et al.*, 2020] as our transformer-based encoder-decoder architecture. In this way, the general text generation knowledge can be directly reused.

3.3 Decoding with Opinion Constraints

In this study, we employ a tire-based constrained decoding algorithm [Chen *et al.*, 2020; Lu *et al.*, 2021] for generating the linearized opinion tree token-by-token.

During constrained decoding, the opinion schema knowledge is injected as the prompt of the decoder and ensures the generation of a valid opinion tree. Concretely, our constrained decoding method dynamically chooses and prunes a candidate vocabulary V_t based on the current generated state. Specifically, each generation step has three kinds of candidate vocabulary V_t: *opinion schema*: label names of category and polarity are predefined; *mention opinion string*: aspect term and opinion word are the text span in the raw input; *structure indicator*: "(" and ")" are used to combine opinion schema and mention opinion strings.

At the generation step t, the candidate vocabulary V_t is the children nodes of the last generated node. For instance, at the generation step with the string "(" to be generated, the candidate vocabulary V_t is {"(",")"}. When generating the aspect category, polarity, and text span, the decoding process can be considered as executing a search on the sub-tree of the tire tree. For example, the candidate vocabulary V_t for polarity is {"Positive", "Negative"}.

3.4 Objective Functions and Training

In this subsection, we show the objective functions and training process of the proposed model.

The goal is to maximize the output linearized opinion tree X_T probability given the review sentence X_O. Therefore, we optimize the negative log-likelihood loss function:

$$\mathcal{L} = -\frac{1}{|\tau|} \sum_{(X_O, X_T) \in \tau} \log p(X_T|X_O; \theta) \tag{4}$$

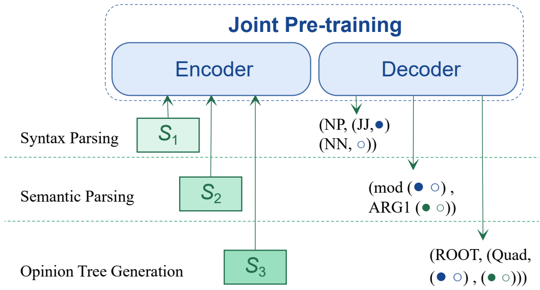


Figure 3: Joint pre-training model.

where θ is the model parameters, and (X_O, X_T) is a (*sentence, tree*) pair in training set τ, then

$$\log p(X_T|X_O;\theta) = \\ = \sum_{i=1}^{n} \log p(x_T^i|x_T^1, x_T^2, ...x_T^{i-1}, X_O;\theta) \quad (5)$$

where $p(x_T^i|x_T^1, x_T^2, ...x_T^{i-1}, X_O;\theta)$ is calculated by the decoder.

4 Joint Pre-training

In this study, we explore joint learning of several pre-training tasks to integrate syntax and semantic features and optimize for the performance of opinion tree generation. These pre-trained models can capture linguistic features from different perspectives and are very effective for forming the opinion tree structure. To build these pre-trained models, we explore three different yet relevant sequence-to-sequence (seq2seq) tasks. Here, syntax parsing is used to capture the syntax features, and semantic parsing is used to capture the semantic features. It is worth noting that the pre-training task of opinion tree generation is in a similar spirit of self-training.

To be consistent with the seq2seq model for opinion tree generation, the pre-trained models are all built on the Transformer [Vaswani *et al.*, 2017]. As shown in Figure 3, for each pre-training task, we learn a seq2seq model which will be used to initialize the seq2seq model for opinion tree generation in the fine-tuning phase. When building the pre-trained models, we merge all of these pre-training tasks and construct a shared vocabulary. In addition, we add a unique preceding tag to the target side of training data to distinguish the task of each training instance. In the following, we will illustrate all the pre-training tasks, and also discuss the fine-tuning method.

4.1 Syntax Parsing

In this study, we treat syntax parsing as a seq2seq constituent parsing model [Choe and Charniak, 2016]. Building such a model requires a training dataset that consists of sentences with constituency parse trees. To construct a silver treebank, we parse the sentences using an off-the-shelf parser[1]. Then, we linearize the automatic parse trees to get syntax sequences. Finally, we train a seq2seq-based syntax parsing model on the silver corpus that will be used as a pre-trained model.

4.2 Semantic Parsing

Semantic parsing is a seq2seq Abstract Meaning Representation (AMR) parsing model [Banarescu *et al.*, 2013; Cai and Lam, 2020]. AMR parsing aims to translate a textual sentence into a directed and acyclic graph which consists of concept nodes, and edges representing the semantic relations between the nodes. In this study, the semantic parsing is trained on a silver corpus of auto-parsed AMR graphs [Xu *et al.*, 2020a]. To construct such a corpus, we apply an off-the-shelf parser[2] of AMR parsing to process the sentences. Then, we adopt the linearization process to obtain the AMR sequence. Finally, we train the semantic parsing pre-trained model on the silver corpus.

4.3 Opinion Tree Generation

The pre-training task of opinion tree generation is in a similar spirit of self-training. In this study, opinion tree generation is a seq2seq model trained on a silver corpus of auto-parsed opinion trees. To construct such a corpus, we apply the baseline system of opinion tree generation to process the sentences. Then, we adopt the linearization process illustrated in Figure 2 to obtain linearized sequences. Finally, we train a seq2seq-based opinion tree generation model on the silver corpus that will be used as a pre-trained model.

4.4 Fine-tuning

Given a pre-trained model, we propose a multi-task learning strategy [Li and Hoiem, 2018; Xu *et al.*, 2020a] for fine-tuning the opinion tree generation model on a gold corpus. In particular, we use the pre-trained model focused and input sentences of gold opinion tree corpus to generate fine-tuning instances for pre-training tasks. Formally, given an instance s in the gold corpus, and a pre-trained model learned from k pre-training tasks, we first feed the pre-trained model with input s and obtain its k outputs for the k pre-training tasks, respectively. Therefore, each input s in the fine-tuning task is now equipped with $k+1$ outputs, one for the fine-tuning task while the other k for the k pre-training tasks. Meanwhile, each output is associated with a unique preceding tag which indicates the corresponding task.

5 Experiments

In this section, we introduce the datasets used for evaluation and the baseline methods employed for comparison. We then report the experimental results conducted from different perspectives, and analyze the effectiveness of the proposed model with different factors.

5.1 Setting

In this study, we use ACOS dataset [Cai *et al.*, 2021] for our experiments. There are 2,286 sentences in Restaurant domain, and 4,076 sentences in Laptop domain. Following the setting from [Cai *et al.*, 2021], we divide the original dataset

[1] https://stanfordnlp.github.io/CoreNLP/parser-standalone.html

[2] https://github.com/xdqkid/S2S-AMR-Parser

Method	**Restaurant**			**Laptop**		
	P.	R.	F1.	P.	R.	F1.
DP	0.3467	0.1508	0.2104	0.1304	0.0057	0.0800
JET	0.5981	0.2894	0.3901	0.4452	0.1625	0.2381
TAS-BERT	0.2629	0.4629	0.3353	0.4715	0.1922	0.2731
Extract-Classify	0.3854	0.5296	0.4461	0.4556	0.2948	0.3580
BARTABSA	0.5662	0.5535	0.5598	0.4165	0.4046	0.4105
GAS	0.6069	0.5852	0.5959	0.4160	0.4275	0.4217
Paraphrase	0.5898	0.5911	0.5904	0.4177	0.4504	0.4334
Ours	**0.6396**	**0.6174**	**0.6283**	**0.4611**	**0.4479**	**0.4544**

Table 1: Comparison with baselines.

into a training set, a validation set, and a testing set. In addition, we choose 20,000 sentences from Yelp[3], and 20,000 sentences from the laptop domain in Amazon[4] to pre-train the proposed opinion tree generation model with the joint pre-training model.

We employ T5[5] and fine-tune its parameters for our opinion tree generation model. We tune the parameters of our models by grid searching on the validation dataset. We select the best models by early stopping using the Accuracy results on the validation dataset. The dimension of other hidden variables of all the models is 128. The model parameters are optimized by Adam [Kingma and Ba, 2015] with a learning rate of 3e-4. The batch size is 16. Our experiments are carried out with an Nvidia RTX 3090 GPU. The experimental results are obtained by averaging ten runs with random initialization.

In evaluation, a quadruple is viewed as correct if and only if the four elements, as well as their combination, are exactly the same as those in the gold quadruple. On this basis, we calculate the Precision and Recall, and use F1 score as the final evaluation metric for aspect sentiment quadruple extraction [Cai *et al.*, 2021; Zhang *et al.*, 2021a].

5.2 Main Results

We compare the proposed model with various strong baselines on Table 1, where,

- *DP* [Qiu *et al.*, 2011] is a double propagation based method, it is one of the representative rule-based methods for aspect-opinion sentiment triple extraction, we propose to adapt it to our aspect sentiment quadruple extraction task.
- *JET* [Xu *et al.*, 2020b] is an end-to-end framework, it combines the identification of aspects, their corresponding opinions, and their sentiment polarities with a position-aware tagging scheme.
- *TAS-BERT* [Wan *et al.*, 2020] integrates aspect category-based sentiment classification and aspect extraction in a unified framework by attaching the aspect category and the sentiment polarity to the review sentence and using it as the input of BERT.

[3]https://www.yelp.com/dataset

[4]http://jmcauley.ucsd.edu/data/amazon/

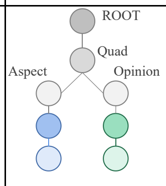
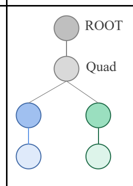
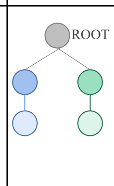
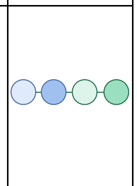
[5]T5_{base}, https://huggingface.co/transformers/model_doc/t5.html

Method	**Restaurant**	**Laptop**
Ours	0.6283	0.4544
-ConsDecoding	0.6249	0.4517
-Pretrained	0.6164	0.4394
-Syntax	0.6206	0.4500
-AMR	0.6159	0.4440
-Opinion	0.6170	0.4414

Table 2: Impact of different factors.

- *Extract-Classify* [Cai *et al.*, 2021] firstly performs aspect-opinion co-extraction, and then predicts category-sentiment given the extracted aspect-opinion pairs.
- *BARTABSA* [Yan *et al.*, 2021] converts all ABSA subtasks into a unified generative formulation, and treats the class index as the target of the generation model.
- *GAS* [Zhang *et al.*, 2021b] tackles all ABSA tasks in a unified generative framework, and formulates ABSA task as a sentiment element sequence generation problem.
- *Paraphrase* [Zhang *et al.*, 2021a] aims to jointly detect all sentiment elements in quads. They propose a paraphrase modeling paradigm to cast the ABSA task to a paraphrase generation process, and joint extract all the sentiment elements.

From the results, we find that generation methods (i.e., BARTABSA, GAS, Paraphrase) give the best performance among the previous systems. It shows that the unified generation architecture can fully utilize the rich label semantics by encoding the natural language label into the target output, and it is very helpful for extracting sentiment elements jointly. In addition, our proposed model outperforms all the previous studies significantly ($p < 0.05$) in all settings, with 3 points of improvement in terms of F1 score on average. This indicates that tree structure is much more helpful than the flat sequence for generating sentiment elements. Furthermore, the results also indicate the effectiveness of joint pre-training model, which is used to integrate semantic structure for opinion tree generation.

ID	(a)	(b)	(c)	(d)
Tree				
Linearization	(ROOT, (Quad (Aspect, (● ○)) , (Opinion, (● ○))))	(ROOT, (Quad, (● ○) , (● ○)))	(ROOT, (● ○) , (● ○))	(●, ○ , ● ,○)
Restaurant	**0.6283**	0.6023	0.5995	0.5959
Laptop	**0.4544**	0.4370	0.4314	0.4217

aspect term　opinion word
aspect category　polarity

Figure 4: Results of different tree structures.

Figure 5: Influence of pre-trained data size.

5.3 Impact of Different Factors

As shown in Table 2, we then employ ablation experiments to analyze the impact of different factors in the proposed model. If we remove the constrained decoding stage (-ConsDecoding) of the tree generation model, the performance drops to 0.6249 and 0.4517 respectively. It indicates that the opinion schema knowledge is very important for ensuring the generation of a valid opinion tree.

In addition, if we totally remove the joint pre-training tasks (-Pretrained) of the proposed model, the performance drops to 0.6164 and 0.4394 respectively. It indicates that the joint pre-training tasks which integrate syntax and semantic features help optimize the performance of opinion tree generation. Furthermore, we also find that all the pre-training tasks are beneficial to generate the opinion tree. If we remove one of these tasks, the performance will be lower than our proposed model.

6 Analysis and Discussion

In this section, we give some analysis and discussion to show the importance of the opinion tree structure and the joint pre-training tasks.

6.1 Analysis of Different Tree Structure

We firstly analyze the influence of different tree structures. As shown in Figure 4, (a) is the proposed tree structure, the detailed discussion can be found in Section 3.1; (b) is similar with (a), but aspect category and opinion polarity are connected with the Quad node directly; (c) is a simple tree which employs a root node to connect each (aspect term, category) or (opinion word, polarity) pair; (d) is a flat sequence which has been studied in previous researches [Yan *et al.*, 2021; Zhang *et al.*, 2021b]. Note that, we linearize all of these tree structures, and employ T5 as the backbone for training these tree/sequence generation models.

From the results in Figure 4, we find that all the tree generation models outperform the flat sequence generation model. It shows that the tree structure is beneficial to capture the semantic relations between aspect terms and opinion words, and it is much more helpful for generating sentiment elements than the sequence based generation model. Furthermore, we find that the proposed tree structure (a) is more effective than the simple tree structure (c). It indicates that the proposed tree structure can capture more semantic structure information.

6.2 Influence of Pre-trained Data Size

The pre-training tasks can be used to integrate syntax and semantic features, and they are very important for forming the tree structure. We thus analyze the data size for these pre-training tasks in Figure 5. From the figure, we find that the more training data, the higher performance our proposed model can reach with the average F1-score in the two domains. It indicates that the large-scale unlabeled data is very useful for training the pre-training tasks. In addition, it also indicates that our proposed joint pre-training model with large-scale unlabeled data is very important for the opinion tree generation task.

7 Conclusion

In this study, we propose a new task called opinion tree generation, which aims to jointly detect all sentiment elements in a tree for a given review sentence. Specifically, we employ a sequence-to-sequence architecture to generate the opinion tree. Furthermore, we design two strategies for effective sequence-to-sequence opinion tree generation. Firstly, we propose a constrained decoding algorithm, which can guide the generation process using opinion schemas. In this way, the opinion knowledge can be injected and exploited during inference on-the-fly. Secondly, we explore joint learning of several pre-training tasks to integrate syntax and semantic features, and optimize for the performance of opinion tree generation. Experimental results show that our proposed model can achieve state-of-the-art performance in ABSA. In addition, the results also validate the strong generality of the proposed framework which can be easily adapted to arbitrary ABSA tasks without additional task-specific model design.

Acknowledgments

We would like to thank Prof. Junhui Li and Prof. Guodong Zhou for their helpful advice and discussion during this work. Also, we would like to thank the anonymous reviewers for their insightful and valuable comments.

This work is supported by the National Natural Science Foundation of China (No. 62076175, No. 61976146), and the Jiangsu Innovation Doctor Plan.

References

[Banarescu *et al.*, 2013] Laura Banarescu, Claire Bonial, Shu Cai, Madalina Georgescu, Kira Griffitt, Ulf Hermjakob, Kevin Knight, Philipp Koehn, Martha Palmer, and Nathan Schneider. Abstract meaning representation for sembanking. In *LAW-ID@ACL 2013*, pages 178–186, 2013.

[Bu *et al.*, 2021] Jiahao Bu, Lei Ren, Shuang Zheng, Yang Yang, Jingang Wang, Fuzheng Zhang, and Wei Wu. ASAP: A chinese review dataset towards aspect category sentiment analysis and rating prediction. In *NAACL-HLT 2021*, pages 2069–2079, 2021.

[Cai and Lam, 2020] Deng Cai and Wai Lam. AMR parsing via graph-sequence iterative inference. In *ACL 2020*, pages 1290–1301, 2020.

[Cai *et al.*, 2020] Hongjie Cai, Yaofeng Tu, Xiangsheng Zhou, Jianfei Yu, and Rui Xia. Aspect-category based sentiment analysis with hierarchical graph convolutional network. In *COLING 2020*, pages 833–843, 2020.

[Cai *et al.*, 2021] Hongjie Cai, Rui Xia, and Jianfei Yu. Aspect-category-opinion-sentiment quadruple extraction with implicit aspects and opinions. In *ACL/IJCNLP 2021*, pages 340–350, 2021.

[Chen *et al.*, 2020] Pinzhen Chen, Nikolay Bogoychev, Kenneth Heafield, and Faheem Kirefu. Parallel sentence mining by constrained decoding. In *ACL 2020*, pages 1672–1678, 2020.

[Choe and Charniak, 2016] Do Kook Choe and Eugene Charniak. Parsing as language modeling. In *EMNLP 2016*, pages 2331–2336, 2016.

[Kingma and Ba, 2015] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *ICLR 2015*, 2015.

[Li and Hoiem, 2018] Zhizhong Li and Derek Hoiem. Learning without forgetting. *IEEE Trans. Pattern Anal. Mach. Intell.*, 40(12):2935–2947, 2018.

[Lu *et al.*, 2021] Yaojie Lu, Hongyu Lin, Jin Xu, Xianpei Han, Jialong Tang, Annan Li, Le Sun, Meng Liao, and Shaoyi Chen. Text2event: Controllable sequence-to-structure generation for end-to-end event extraction. In *ACL/IJCNLP 2021*, pages 2795–2806, 2021.

[Peng *et al.*, 2020] Haiyun Peng, Lu Xu, Lidong Bing, Fei Huang, Wei Lu, and Luo Si. Knowing what, how and why: A near complete solution for aspect-based sentiment analysis. In *AAAI 2020*, pages 8600–8607, 2020.

[Qiu *et al.*, 2011] Guang Qiu, Bing Liu, Jiajun Bu, and Chun Chen. Opinion word expansion and target extraction through double propagation. *Comput. Linguistics*, 37(1):9–27, 2011.

[Raffel *et al.*, 2020] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67, 2020.

[Tang *et al.*, 2016] Duyu Tang, Bing Qin, Xiaocheng Feng, and Ting Liu. Effective lstms for target-dependent sentiment classification. In *COLING 2016*, pages 3298–3307, 2016.

[van Noord and Bos, 2017] Rik van Noord and Johan Bos. Neural semantic parsing by character-based translation: Experiments with abstract meaning representations. *CoRR*, abs/1705.09980, 2017.

[Vaswani *et al.*, 2017] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NIPS 2017*, pages 5998–6008, 2017.

[Vinyals *et al.*, 2015] Oriol Vinyals, Lukasz Kaiser, Terry Koo, Slav Petrov, Ilya Sutskever, and Geoffrey E. Hinton. Grammar as a foreign language. In *NIPS 2015*, pages 2773–2781, 2015.

[Wan *et al.*, 2020] Hai Wan, Yufei Yang, Jianfeng Du, Yanan Liu, Kunxun Qi, and Jeff Z. Pan. Target-aspect-sentiment joint detection for aspect-based sentiment analysis. In *AAAI 2020*, pages 9122–9129, 2020.

[Wang *et al.*, 2020] Kai Wang, Weizhou Shen, Yunyi Yang, Xiaojun Quan, and Rui Wang. Relational graph attention network for aspect-based sentiment analysis. In *ACL 2020*, pages 3229–3238, 2020.

[Xu *et al.*, 2020a] Dongqin Xu, Junhui Li, Muhua Zhu, Min Zhang, and Guodong Zhou. Improving AMR parsing with sequence-to-sequence pre-training. In *EMNLP 2020*, pages 2501–2511, 2020.

[Xu *et al.*, 2020b] Lu Xu, Hao Li, Wei Lu, and Lidong Bing. Position-aware tagging for aspect sentiment triplet extraction. In *EMNLP 2020*, pages 2339–2349, 2020.

[Xue and Li, 2018] Wei Xue and Tao Li. Aspect based sentiment analysis with gated convolutional networks. In *ACL 2018*, pages 2514–2523, 2018.

[Yan *et al.*, 2021] Hang Yan, Junqi Dai, Tuo Ji, Xipeng Qiu, and Zheng Zhang. A unified generative framework for aspect-based sentiment analysis. In *ACL/IJCNLP 2021*, pages 2416–2429, 2021.

[Zhang and Qian, 2020] Mi Zhang and Tieyun Qian. Convolution over hierarchical syntactic and lexical graphs for aspect level sentiment analysis. In *EMNLP 2020*, pages 3540–3549, 2020.

[Zhang *et al.*, 2021a] Wenxuan Zhang, Yang Deng, Xin Li, Yifei Yuan, Lidong Bing, and Wai Lam. Aspect sentiment quad prediction as paraphrase generation. In *EMNLP 2021*, pages 9209–9219, 2021.

[Zhang *et al.*, 2021b] Wenxuan Zhang, Xin Li, Yang Deng, Lidong Bing, and Wai Lam. Towards generative aspect-based sentiment analysis. In *ACL/IJCNLP 2021*, pages 504–510, 2021.

[Zhou *et al.*, 2021] Jiawei Zhou, Tahira Naseem, Ramón Fernandez Astudillo, Young-Suk Lee, Radu Florian, and Salim Roukos. Structure-aware fine-tuning of sequence-to-sequence transformers for transition-based AMR parsing. In *EMNLP 2021*, pages 6279–6290, 2021.