Opinion Tree Parsing for Aspect-based Sentiment Analysis

Xiaoyi Bao[1], Xiaotong Jiang[1], Zhongqing Wang[1,*], Yue Zhang[2,3], and Guodong Zhou[1]
[1]Natural Language Processing Lab, Soochow University, Suzhou, China
[2]School of Engineering, Westlake University
[3]Institute of Advanced Technology, Westlake Institute for Advanced Study
p2213545413@outlook.com,devjiang@outlook.com
yue.zhang@wias.org.cn
{wangzq,gdzhou}@suda.edu.cn

Abstract

Extracting sentiment elements using pre-trained generative models has recently led to large improvements in aspect-based sentiment analysis benchmarks. However, these models always need large-scale computing resources, and they also ignore explicit modeling of structure between sentiment elements. To address these challenges, we propose an opinion tree parsing model, aiming to parse all the sentiment elements from an opinion tree, which is much faster, and can explicitly reveal a more comprehensive and complete aspect-level sentiment structure. In particular, we first introduce a novel context-free opinion grammar to normalize the opinion tree structure. We then employ a neural chart-based opinion tree parser to fully explore the correlations among sentiment elements and parse them into an opinion tree structure. Extensive experiments show the superiority of our proposed model and the capacity of the opinion tree parser with the proposed context-free opinion grammar. More importantly, the results also prove that our model is much faster than previous models. Our code can be found in https://github.com/HoraceXIaoyiBao/OTP4ABSA-ACL2023.

1 Introduction

Aspect-based sentiment analysis (ABSA) has drawn increasing attention in the community, which includes four subtasks: aspect term extraction, opinion term extraction, aspect term category classification and aspect-level sentiment classification. The first two subtasks aim to extract the aspect term and the opinion term appearing in one sentence. The goals of the remaining two subtasks are to detect the category and sentiment polarity towards the extracted aspect term.

Previously, most ABSA tasks are formulated as either sequence-level (Qiu et al., 2011; Peng et al., 2020; Cai et al., 2021) or token-level classification problems (Tang et al., 2016). However, these methods usually suffer severely from error propagation because the overall prediction performance hinges on the accuracy of every step (Peng et al., 2020). Therefore, recent studies tackle the ABSA problem with 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,a) as the target of generation model. More recently, Bao et al. (2022) addresses the importance of correlations among sentiment elements (e.g., aspect term, opinion term), and proposes an opinion tree generation model, which aims to jointly detect all sentiment elements in a tree structure.

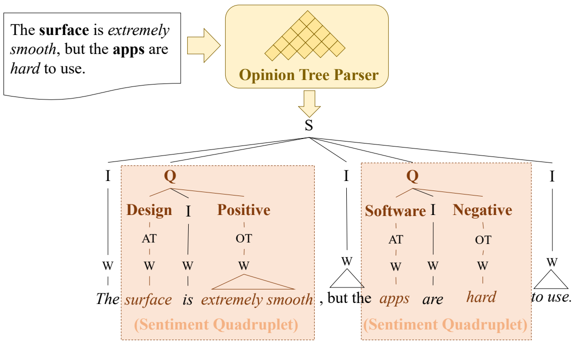

Figure 1: Example of opinion tree parsing.

The major weakness of generative approaches is the training and inference efficiency, they always need large-scale computing resources. In addition, these generative approaches lack certain desirable properties. There are no structural guarantees of structure well-formedness, i.e. the model may predict strings that can not be decoded into valid opinion trees, and post-processing is required. Furthermore, predicting linearizations ignores the implicit alignments among sentiment elements, which provide a strong inductive bias.

As shown in Figure 1, we convert all the sentiment elements into an opinion tree and design a neural chart-based *opinion tree parser* to address

[*] Corresponding author

Findings of the Association for Computational Linguistics: ACL 2023, pages 7971–7984
July 9-14, 2023 ©2023 Association for Computational Linguistics

these shortcomings. The opinion tree parser is much simpler and faster than generative models. It scores each span independently and performs a global search over all possible trees to find the highest-score opinion tree (Kitaev and Klein, 2018; Kitaev et al., 2019). It explicitly models tree structural constraints through span-based searching and yield alignments by construction, thus guaranteeing tree structure well-formedness.

One challenge to the above is that not all the review texts contain standard sentiment quadruplets (i.e., aspect term, opinion term, aspect category, and polarity) which can be easily formed in an opinion tree (Bao et al., 2022). For example, there may be more than one opinion term correlated with an aspect term and vice versa. In addition, aspect or opinion terms might be implicit. According to our statistics, such irregular situations appear in more than half of review texts. In this study, we propose a novel *context-free opinion grammar* to tackle these challenges. The grammar is generalized and well-designed, it is used to normalize the sentiment elements into a comprehensive and complete opinion tree. Furthermore, it contains four kinds of conditional rules, i.e., one-to-many, mono-implicit, bi-implicit, cross-mapping, which are used to solve the irregular situations in opinion tree parsing.

The detailed evaluation shows that our model significantly advances the state-of-the-art performance on several benchmark datasets. In addition, the empirical studies also indicate that the proposed opinion tree parser with context-free opinion grammar is more effective in capturing the sentiment structure than generative models. More importantly, our model is much faster than previous models.

2 Related Work

As a complex and challenging task, aspect-based sentiment analysis (ABSA) consists of numerous sub-tasks. The researches on ABSA generally follow a route from handling single sub-task to complex compositions of them. The fundamental subtasks focus on the prediction of a single sentiment element, such as extracting the aspect term (Qiu et al., 2011; Tang et al., 2016; Wang et al., 2021), detecting the mentioned aspect category (Bu et al., 2021; Hu et al., 2019), and predicting the sentiment polarity for a given aspect (Tang et al., 2016; Chen et al., 2022a; Liu et al., 2021; Seoh et al., 2021; Zhang et al., 2022).

Since the sentiment elements are natural correlated, many studies focus on exploring the joint extraction of pairwise sentiment elements, including aspect and opinion term extraction (Xu et al., 2020; Li et al., 2022); aspect term extraction and its polarity detection (Zhang and Qian, 2020); aspect category and polarity detection (Cai et al., 2020). Furthermore, recent studies also employed end-to-end models to extract all the sentiment elements in triplet or quadruple format (Peng et al., 2020; Wan et al., 2020; Cai et al., 2021; Zhang et al., 2021a; Chen et al., 2022b; Mukherjee et al., 2021).

More recently, studies using pre-trained encoder-decoder language models show great improvements in ABSA (Zhang et al., 2021a). They either treated the class index (Yan et al., 2021) or the desired sentiment element sequence (Zhang et al., 2021b) as the target of the generation model. in addition, Bao et al. (2022) addressed the importance of correlations among sentiment elements, and proposed an opinion tree generation model, which aims to jointly detect all sentiment elements in a tree structure. However, the generative models always need large-scale computing resources, they also cannot guarantee the structure well-formedness, and ignores the implicit alignments among sentiment elements.

In this study, we propose a novel opinion tree parser, which aims to model and parse the sentiment elements from the opinion tree structure. The proposed model shows significant advantages in both decoding efficiency and performance as it is much faster and more effective in capturing the sentiment structure than generative models. Furthermore, we design a context-free opinion grammar to normalize the opinion tree structure, and improve parser's applicability decisions for complex compounding phenomena.

3 Overview of Proposed Model

Aspect-based sentiment analysis aims to extract all kinds of sentiment elements and their relations from review text. Basically, there are four kinds of sentiment elements in the review text: *aspect term* denotes an entity and its aspect indicating the opinion target, which is normally a word or phrase in the text; *aspect category* represents a unique predefined category for the aspect in a particular domain; *opinion term* refers the subjective statement on an aspect, which is normally a subjective word or phrase in the text; *polarity* is the prede-

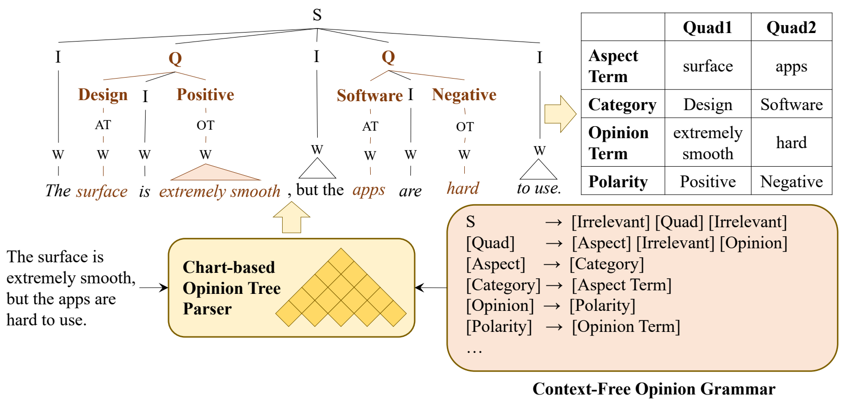
S
I
Q
I
Q
I
Design
I
Positive
Software
I
Negative
AT
OT
AT
OT
W
W
W
W
W
W
W
W
W
The surface is extremely smooth , but the apps are hard to use.
Quad1
Quad2
Aspect Term
surface
apps
Category
Design
Software
Opinion Term
extremely smooth
hard
Polarity
Positive
Negative
The surface is extremely smooth, but the apps are hard to use.
Chart-based Opinion Tree Parser
S → [Irrelevant] [Quad] [Irrelevant]
[Quad] → [Aspect] [Irrelevant] [Opinion]
[Aspect] → [Category]
[Category] → [Aspect Term]
[Opinion] → [Polarity]
[Polarity] → [Opinion Term]
…
Context-Free Opinion Grammar

Figure 2: Overview of proposed model.

fined semantic orientation (e.g., positive, negative, or neutral) toward the aspect.

As shown in Figure 2, we convert all the sentiment elements into an opinion tree, and we design a chart-based opinion tree parser with context-free opinion grammar to parse the opinion tree from review text. In particular, we firstly propose a context-free opinion grammar to normalize the sentiment elements into an opinion tree. We then perform a neural chart-based opinion tree parser to parse the opinion tree structure from a given review text. Since all the sentiment elements are normalized into the opinion tree, it is easy to recover them from the tree. In the next two sections, we will discuss the context-free opinion grammar and the opinion tree parser in detail.

Name	Description
Q	Quad of sentiment elements
I	Irrelevant content (*e.g., the, but, are*)
A	Aspect pair (*Category, Aspect Term*)
O	Opinion pair (*Polarity, Opinion Term*)
C	Category of aspect (*e.g., Design, Software*)
P	Polarity towards the aspect (*Positive, Negative*)
AT	Aspect term (*e.g., surface, apps*)
OT	Opinion term (*e.g., smooth, hard*)
W	Word

Table 1: Notation of non-terminals.

4 Context-Free Opinion Grammar

In this study, we propose a novel context-free opinion grammar to normalize the opinion tree structure. In the below of this section, we first introduce basic definitions of context-free opinion grammar. After that, we give some conditional rules to solve irregular situations and show some examples to illustrate the effectiveness of proposed grammar.

4.1 Basic Definitions

A context-free opinion grammar (CFOG) is a tuple $G = (N, \Sigma, P, S)$, where N and Σ are finite, disjoint sets of non-terminal and terminal symbols, respectively, Table 1 gives the notation of non-terminals. $S \in N$ is the start symbol and P is a finite set of rules. Each rule has the form $A \rightarrow \alpha$, where $A \in N$, $\alpha \in V_I^*$ and $V_I = N \cup \Sigma$.

The top of Figure 2 gives an example of opinion parsing tree. Each terminal in the tree is either an irrelevant word or a sentiment element like aspect or opinion term. Each non-terminal combines terminals or non-terminals to create a sub-tree of sentiment elements. In order to make the description as clear as possible, we begin with the basic rules allowed by our grammar:

S → I Q I // S → irrelevant content,quad,irrelevant content

Q → A I O | O I A | ϵ // quad → (aspect,opinion) or (opinion,aspect)

Q → Q I Q // multiple quads

A → C // aspect → category

C → AT // category → aspect term

O → P // opinion → polarity

P → OT // polarity → opinion term

AT → W // aspect term → word

OT → W // opinion term → word

I → W // irrelevant content → word

W→ W W | ϵ

W → happy | to | great | party | but | have | ...

C ⇔ Surface | Laptop | ... // C is replaced with a certain category

P ⇔ Positive | Negative | Neutral // P is replaced with a certain polarity

In the above notations, the rules bring out the

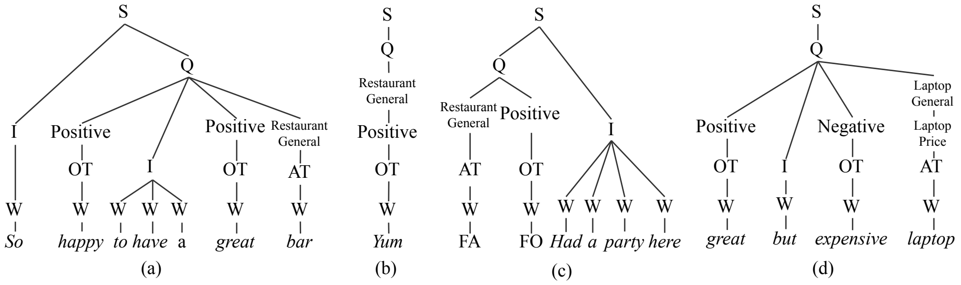

Figure 3: Examples of opinion trees with conditional rules and pruning approach.

grammatical relations among the elements of a standard sentiment quadruplet. For example, I is used to define the irrelevant content in the review sentence, and Q is used to describe a sentiment quadruple. In addition, the components of quadruple, i.e., A and O, are used to denote the aspect pair (category C and aspect term AT) and opinion pair (polarity P and opinion term OT). Since the opinion trees built under the above grammar may be too complicated, we adopt a pruning approach to reduce the duplication in the trees, detail discussion of pruning can be found in Appendix A.

4.2 Conditional Rules

Although the basic rules can be used to parse an opinion tree with standard quadruplets, they cannot handle irregular situations. In this subsection, we introduce conditional rules to improve rule applicability for complex compounding phenomena.

One-to-Many means that there is more than one opinion term correlated with an aspect term, and vice versa. For example, in the review sentence "So *happy* to have a *great bar*", both opinion terms "*happy*" and "*great*" are mapped to the same aspect term "*bar*". In this study, we attach successor elements to the preceding one and charge the rule A and O below for solving this situation:

// multiple aspects map to one opinion
A → A I A
// multiple opinions map to one aspect
O → O I O

Then, the above cause can be correctly parsed through these two new rules. The example of parsing result is shown in Figure 3(a).

Mono-Implicit means that either aspect term or opinion term is missing in the review text. Given a review sentence "*Yum*", only an opinion term appears in the sentence. For solving this problem, we attach the opinion to corresponding aspect node or attach the aspect to corresponding opinion node:

// implicit aspect term
Q → C; C → O   // quad →category→opinion
// implicit opinion term
Q → P; P → A   // quad →polarity → aspect

An example of this solution can be found in Figure 3(b).

Bi-Implicit denotes that both the aspect term and opinion term are missing in the review text. As shown in the review sentence "Had a party here", although we know that the authors express a positive opinion, both aspect term and opinion term do not appear in the sentence. To solve the situation, we insert two fake tokens FA and FO at the beginning of a sentence as the fake aspect and opinion term. Then, we can use standard rules to parse such sentences with implicit aspect and opinion. Figure 3(c) gives an example of this solution.

Cross-Mapping means that there are more than one aspect category and opinion polarity on the review text, and their correlations are many-to-many. For example, in the review sentence "*Great* but *expensive* laptop", there are two categories "Laptop General" and "Laptop Price" towards the aspect term "laptop". Meanwhile, the opinions towards these two categories are different. The author feels "great" about the "Laptop General", but thinks the "Laptop Price" is "expensive". The solution of such situation is shown in below:

// two categories and two opinion terms towards one aspect term
A → C_1; C_1 → C_2; C_2 → AT
// two categories and two opinion terms towards one opinion term
O → P_1; P_1 → P_2; P_2 → OT

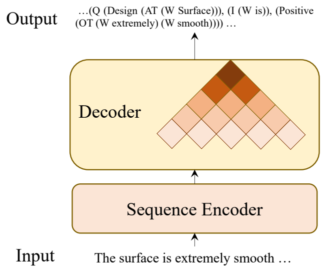
Output
…(Q (Design (AT (W Surface))), (I (W is)), (Positive (OT (W extremely) (W smooth)))) …
Decoder
Sequence Encoder
Input
The surface is extremely smooth …

Figure 4: Neural chart-based pinion tree parser.

Then, we use the shortest path to detect the correlation between aspect category and opinion term. As shown in Figure 3(d), since the distance between "Laptop General" and "great" is shorter than " *expensive* ", we connect "Laptop General" with " *great*", and then connect "Laptop Price" with " *expensive* ".

In summary, based on the basic and conditional rules, the proposed context-free opinion grammar can solve most situations in aspect-based sentiment analysis, and would help parse a comprehensive and complete opinion tree.

5 Opinion Tree Parser

In this study, we employ a neural chart-based opinion tree parser to parse sentiment elements from the opinion tree structure. As shown in Figure 4, the opinion tree parser follows an encoder-decoder architecture (Kitaev and Klein, 2018; Kitaev et al., 2019; Cui et al., 2022). It scores each span independently and performs a global search over all possible trees to find the highest-score opinion tree. In particular, the process of opinion tree parsing can be separated into two stages: context-aware encoding and chart-based decoding, we will discuss these in the below subsections.

5.1 Span Scores and Context-Aware Encoding

Given a review text $X = \{x_1, ..., x_n\}$, its corresponding opinion parse tree T is composed by a set of labeled spans:

$$T = \{(i_t, j_t, l_t)\}|_{t=1}^{|T|} \tag{1}$$

where i_t and j_t represent the t-th span's fencepost positions and l_t represents the span label.

We use a self-attentive encoder as the scoring function $s(i, j)$, and a chart decoder to perform a global-optimal search over all possible trees to find the highest-scoring tree given the review text. In particular, given an input review text $X = \{x_1, ..., x_n\}$, a list of hidden representations $H_1^n = \{h_1, h_2, ..., h_n\}$ is produced by the encoder, where h_i is a hidden representation of the input token x_i. The representation of a span (i, j) is constructed by:

$$v_{i,j} = h_j - h_i \tag{2}$$

Finally, $v_{i,j}$ is fed into an MLP to produce real valued scores $s(i, j,)$ for all labels:

$$s(i, j) = W_2 ReLU(W_1 v_{i,j} + b_1) + b_2 \tag{3}$$

where W_1, W_2, b_1 and b_2 are trainable parameters, $W_2 \in R^{|H| \times |L|}$ can be considered as the label embedding matrix, where each column in W_2 corresponds to the embedding of a particular constituent label. $|H|$ represents the hidden dimension and $|L|$ is the size of the label set.

5.2 Tree Scores and Chart-based Decoding

The model assigns a score $s(T)$ to each tree T, which can be decomposed as:

$$s(T) = \sum_{(i,j,l) \in T} s(i, j, l) \tag{4}$$

At test time, the model-optimal tree can be found efficiently using a CKY-style inference algorithm. Given the correct tree $T*$, the model is trained to satisfy the margin constraints:

$$s(T*) \geq s(T) + \Delta(T, T^*) \tag{5}$$

for all trees T by minimizing the hinge loss:

$$max(0, \underset{T \neq T^*}{max} [s(T) + \Delta(T, T^*)] - s(T^*)) \tag{6}$$

Here Δ is the Hamming loss on labeled spans, and the tree corresponding to the most-violated constraint can be found using a slight modification of the inference algorithm used at test time.

6 Experiments

In this section, we introduce the dataset used for evaluation and the baseline methods employed for comparison. We then report the experimental results conducted from different perspectives.

6.1 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.

Method	**Restaurant**			**Laptop**		
	P.	R.	F1.	P.	R.	F1.
BERT-CRF	0.3717	0.3055	0.3353	0.2966	0.2562	0.2749
JET	0.5731	0.2754	0.3720	0.4326	0.1435	0.2155
TAS-BERT	0.2611	0.4509	0.3307	**0.4654**	0.1892	0.2690
Extract-Classify	0.3812	0.5144	0.4378	0.4523	0.2822	0.3475
BARTABSA	0.5793	0.5513	0.5650	0.4032	0.3853	0.3940
GAS	0.5871	0.5694	0.5781	0.3989	0.3917	0.3953
Paraphrase	0.5977	**0.6045**	0.6011	0.3842	**0.3930**	0.3885
OTG	0.6094	0.5988	0.6040	0.4102	0.3901	0.3998
Ours	**0.7113**	0.5608	**0.6271**	0.4512	0.3791	**0.4120**

Table 2: Comparison with baselines.

Domain	**Train**	**Validation**	**Test**
Restaurant	1,529	171	582
Laptop	2,929	326	816

Table 3: Distribution of the dataset.

(2021), we divide the original dataset into a training set, a validation set, and a testing set. In particular, we remove some sentences (1.5% among all the sentences) which cannot be parsed (e.g., one-to-many with implicit term, nested, overlapped). The distribution of the dataset can be found in Table 3.

We tune the parameters of our models by grid searching on the validation dataset. For fair comparison, we employ T5 (Raffel et al., 2020) and fine-tune its parameters not only for our opinion tree parser's encoder, but also for the backbone of all other generative methods. The model parameters are optimized by Adam (Kingma and Ba, 2015) with a learning rate of 5e-5. The batch size is 128 with a maximum 512 token length. Our experiments are carried out with a 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).

6.2 Main Results

We compare the proposed opinion tree parser with several classification-based aspect-based sentiment analysis models, including, *BERT-CRF* (Devlin et al., 2019), *JET* (Xu et al., 2020), *TAS-BERT* (Wan et al., 2020) and *Extract-Classify* (Cai et al., 2021). In addition, generative models are also compared, such as *BARTABSA* (Yan et al., 2021), *GAS* (Zhang et al., 2021b), *Paraphrase* (Zhang et al., 2021a) and *OTG* (Bao et al., 2022).[1]

As shown in Table 2, we find that generative models give the best performance among the previous systems. It shows that the unified generation architecture helps extract sentiment elements jointly. Meanwhile, our proposed model outperforms all the previous studies significantly ($p < 0.05$) in all settings. It indicates that the chart-based opinion parser is more useful for explicitly modeling tree structural constraints, while previous generative models cannot guarantee the structure well-formedness, and their generated linearized string ignores the implicit alignments among sentiment elements. Furthermore, the results also indicate the effectiveness of the context-free opinion grammar, which is used to form the sentiment structure into an opinion tree.

6.3 Comparison of Decoding Efficiency

Table 4 compares different models in terms of decoding speed. For a fair comparison, we re-run all previous models on the same GPU environment. The results are averaged over 3 runs. In addition, the settings of batch size are the same for all the models.

As we can see, for generative models (Zhang et al., 2021b,a; Bao et al., 2022), they have to generate words one by one, leading to their low speed, and the beam searching during decoding makes the speed much slower. Meanwhile, based

[1] The implementations of JET, TAS-BERT, Extract-Classify and OTG are based on their official codes, we re-implement the remaining by ourselves.

Method	Encoder	Time (s)
BERT-CRF	BERT	1.96
JET		2.83
Ours		**0.81**
GAS	T5	58.2
Paraphrase		61.3
OTG		64.9
Ours		**1.04**

Table 4: Decoding efficiency of different models.

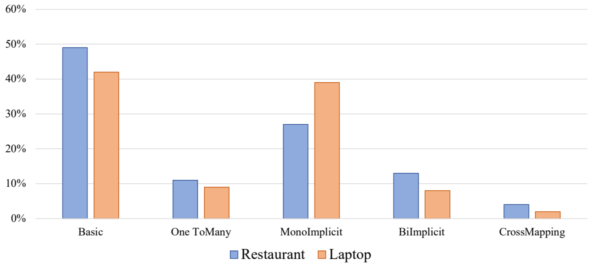

Figure 5: Statistic of regular and irregular situations of opinion trees.

Rules	Restaurant	Laptop
Basic	0.4558	0.2727
+OneToMany	0.5812	0.3175
+MonoImplicit	0.4856	0.3632
+BiImplicit	0.5167	0.2984
+CrossMapping	0.4598	0.2786
Ours	0.6271	0.4120

Table 5: Results of different conditional rules in context-free opinion grammar with F1-score measurement.

on span-based searching, our chart-based opinion tree parser achieves a much higher speed. In addition, the speed of proposed opinion tree parser is faster than the classification-based models (e.g., BERT-CRF, JET). It may be due to that these classification-based models extract the sentiment elements one by one as pipeline systems. It also indicates the effectiveness of the chart-based parser and span-based searching, which could parallelly extract the sentiment elements in the sentence.

7 Analysis and Discussion

In this section, we give some analysis and discussion to show the effectiveness of proposed opinion tree parser for aspect-based sentiment analysis.

Method	Restaurant	Laptop
BERT-CRF	0.3353	0.2749
Zhang19	0.5021	0.3537
Nguyen21	0.5872	0.3673
Yang22	0.5936	0.3712
Ours	0.6123	0.3748

Table 6: Results of different parsers with F1-score measurement.

7.1 Effect of Context-Free Opinion Grammar

We firstly give the statistic of regular and irregular situations of opinion trees in Figure 5, where $Basic$ is the regular situation which contains full four elements of a quadruple, and others are the irregular situations. From the figure, we find that the distribution of these situations are similar in the two domains: around half of reviews contains regular full quadruple situations, and mono-implicit is the most frequency irregular situations.

We then analyze the effect of different conditional rules which are used to solve irregular situations. As shown in Table 5, we can find that if we only use the basic rules, the performance of opinion tree parser is very low. It may be due to the irregular situations appear in more than half of the review texts. In addition, all the conditional rules are beneficial to parse the opinion tree. Among these rules, one-to-many performs better than others. Furthermore, our proposed model achieves the best performance, which proves the effect of conditional rules.

7.2 Results of Different Tree Parsers

We then analyze the effect of different tree parsers with the proposed context-free opinion tree grammar. In particular, we select three popular parsers which have shown their effect on syntax tree parsing (Zhang et al., 2019; Nguyen et al., 2021) and name entity recognition (Yang and Tu, 2022). Among these parsers, Zhang et al. (2019) is transition-based parser, which constructs a complex output structure holistically, through a state-transition process with incremental output-building actions; Nguyen et al. (2021) and Yang and Tu (2022) are sequence-to-sequence parsers, which employ pointing mechanism for bottom-up parsing and use sequence-to-sequence backbone. For fair comparison, we use RoBERTa-base (Liu et al., 2019) as the backbone of all the parsers and our proposed chart-based opinion tree parser.

Schema	Domain	OTG	Ours
Pair	Restaurant	0.6906	0.7681
	Laptop	0.7201	0.7602
Triple	Restaurant	0.6582	0.7051
	Laptop	0.6562	0.6843
Quad	Restaurant	0.6040	0.6271
	Laptop	0.3998	0.4120

Table 7: Results on different opinion tree schemas with F1-score measurement.

As shown in Table 6, all the parsers outperform the BERT-CRF. It shows the effect of the proposed context-free opinion grammar. No matter which parser we use, it achieves better performance than classification-based models. In addition, our chart-based opinion tree parser outperforms all the other parsers with a remarkable advantage. It may be due to that all the other parsers suffer from error propagation and exposure bias problems. Meanwhile, our proposed chart-based parser could infer parallelly, especially effective in parsing long review texts. Such observation has also been proven in neural constituency parsing (Cui et al., 2022), the chart-based parser reported state-of-the-art performance in that task.

7.3 Impact of Opinion Tree Schemas

We analyze the effect of the proposed model with the opinion tree generation model (OTG) (Bao et al., 2022) in different opinion tree schemas. OTG employs a generative model to jointly detect all sentiment elements in a linearized tree formation with a sequence-to-sequence architecture. In particular, there are three popular schemas: *Pair* means that we only extract aspect term and opinion term from review text (Qiu et al., 2011; Xu et al., 2020; Li et al., 2022), and *Triple* means that we extract aspect term, opinion term, and polarity from review text (Zhang et al., 2021b; Chen et al., 2021). *Quad* is the quadruple schema that extracts the whole four sentiment elements to form the opinion tree (Cai et al., 2020; Zhang et al., 2021a; Bao et al., 2022). Note that, we make minor modifications to the context-free opinion grammar, and let it suitable for Pair and Triple schemas.

From Table 7, we can find that our model outperforms OTG in all the schemas. It indicates that our opinion tree parser model is generalized and can be used to handle different schemas in aspect-based sentiment analysis. It also shows that the parsing strategy is more effect than generative model on capture the structure of sentiment elements. In addition, we also find that the improvement of Pair and Triple are much higher than Quad, it may be due to that the simple schema is easier to normalize and recover.

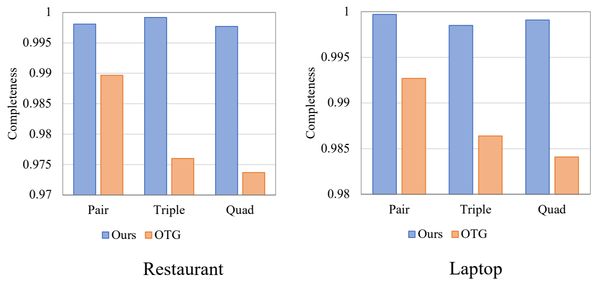

Figure 6: Tree structure completeness of different models.

We then analyze the completeness of the tree structure generated/parsed from OTG and the proposed model. The completeness is calculated through the valid rate of a tree structure. As shown in Figure 6, the completeness of the proposed model is higher than OTG in all the schemas. It shows that our proposed model can explicitly model tree structural constraints, and guarantee tree structure well-formedness. In addition, the high completeness also guarantees the quality of recovery from tree structure to sentiment elements.

Furthermore, case studies in Appendix B are given to make more intuitive comparisons between OTG and proposed opinion tree parser.

8 Conclusion

In this study, we propose a novel opinion tree parsing model, aiming to parse all the sentiment elements into an opinion tree, which can reveal a more comprehensive and complete aspect-level sentiment structure. In particular, we first introduce a novel context-free opinion grammar to normalize the opinion structure. We then employ a neural chart-based opinion tree parser to fully explore the correlations among sentiment elements and parse them in the opinion tree form. Detailed evaluation shows that our model significantly advances the state-of-the-art performance on several benchmarks. The empirical studies also show that the proposed opinion tree parser with context-free opinion grammar is more effective in capturing the opinion tree structure than generative models with a remarkable advantage in computation cost.

9 Limitations

The limitations of our work can be stated from two perspectives. First, the proposed context-free opinion grammar is designed manually. It can be the future work to explore how to automatic generate the grammar. Secondly, we focus on opinion tree parsing in one major language. The performance of other languages remains unknown.

Acknowledgments

We would like to thank Prof. Yue Zhang for his helpful advice and discussion during this work. Also, we would like to thank the anonymous reviewers for their excellent feedback. This work is supported by the China National Key R&D Program (No. 2020AAA0108604), and the National Natural Science Foundation of China (No. 61976180, No. 62006093).

References

Xiaoyi Bao, Zhongqing Wang, Xiaotong Jiang, Rong Xiao, and Shoushan Li. 2022. Aspect-based sentiment analysis with opinion tree generation. In *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence, IJCAI 2022, Vienna, Austria, 23-29 July 2022*, pages 4044–4050. ijcai.org.

Jiahao Bu, Lei Ren, Shuang Zheng, Yang Yang, Jingang Wang, Fuzheng Zhang, and Wei Wu. 2021. ASAP: A Chinese review dataset towards aspect category sentiment analysis and rating prediction. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2069–2079, Online. Association for Computational Linguistics.

Hongjie Cai, Yaofeng Tu, Xiangsheng Zhou, Jianfei Yu, and Rui Xia. 2020. Aspect-category based sentiment analysis with hierarchical graph convolutional network. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 833–843, Barcelona, Spain (Online). International Committee on Computational Linguistics.

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

Chenhua Chen, Zhiyang Teng, Zhongqing Wang, and Yue Zhang. 2022a. Discrete opinion tree induction for aspect-based sentiment analysis. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2051–2064, Dublin, Ireland. Association for Computational Linguistics.

Hao Chen, Zepeng Zhai, Fangxiang Feng, Ruifan Li, and Xiaojie Wang. 2022b. Enhanced multi-channel graph convolutional network for aspect sentiment triplet extraction. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2974–2985, Dublin, Ireland. Association for Computational Linguistics.

Shaowei Chen, Yu Wang, Jie Liu, and Yuelin Wang. 2021. Bidirectional machine reading comprehension for aspect sentiment triplet extraction. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 12666–12674.

Leyang Cui, Sen Yang, and Yue Zhang. 2022. Investigating non-local features for neural constituency parsing. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2022, Dublin, Ireland, May 22-27, 2022*, pages 2065–2075. 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.

Mengting Hu, Shiwan Zhao, Li Zhang, Keke Cai, Zhong Su, Renhong Cheng, and Xiaowei Shen. 2019. CAN: Constrained attention networks for multi-aspect sentiment analysis. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4601–4610, Hong Kong, China. Association for Computational Linguistics.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Nikita Kitaev, Steven Cao, and Dan Klein. 2019. Multilingual constituency parsing with self-attention and pre-training. In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pages 3499–3505. Association for Computational Linguistics.

Nikita Kitaev and Dan Klein. 2018. Constituency parsing with a self-attentive encoder. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, ACL 2018, Melbourne,* *Australia, July 15-20, 2018, Volume 1: Long Papers*, pages 2676–2686. Association for Computational Linguistics.

Junjie Li, Jianfei Yu, and Rui Xia. 2022. Generative cross-domain data augmentation for aspect and opinion co-extraction. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4219–4229, Seattle, United States. Association for Computational Linguistics.

Jian Liu, Zhiyang Teng, Leyang Cui, Hanmeng Liu, and Yue Zhang. 2021. Solving aspect category sentiment analysis as a text generation task. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 4406–4416, Online and Punta Cana, Dominican Republic. 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.

Rajdeep Mukherjee, Tapas Nayak, Yash Butala, Sourangshu Bhattacharya, and Pawan Goyal. 2021. PASTE: A tagging-free decoding framework using pointer networks for aspect sentiment triplet extraction. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 9279–9291, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Thanh-Tung Nguyen, Xuan-Phi Nguyen, Shafiq Joty, and Xiaoli Li. 2021. A conditional splitting framework for efficient constituency parsing. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 5795–5807, Online. Association for Computational Linguistics.

Haiyun Peng, Lu Xu, Lidong Bing, Fei Huang, Wei Lu, and Luo Si. 2020. Knowing what, how and why: A near complete solution for aspect-based sentiment analysis. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 8600–8607.

Guang Qiu, Bing Liu, Jiajun Bu, and Chun Chen. 2011. Opinion Word Expansion and Target Extraction through Double Propagation. *Computational Linguistics*, 37(1):9–27.

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

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

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

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

Qianlong Wang, Zhiyuan Wen, Qin Zhao, Min Yang, and Ruifeng Xu. 2021. Progressive self-training with discriminator for aspect term extraction. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 257–268, Online and Punta Cana, Dominican Republic. 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.

Hang Yan, Junqi Dai, Tuo Ji, Xipeng Qiu, and Zheng Zhang. 2021. A unified generative framework for aspect-based sentiment analysis. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 2416–2429, Online. Association for Computational Linguistics.

Songlin Yang and Kewei Tu. 2022. Bottom-up constituency parsing and nested named entity recognition with pointer networks. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2403–2416, Dublin, Ireland. Association for Computational Linguistics.

Junchi Zhang, Yanxia Qin, Yue Zhang, Mengchi Liu, and Donghong Ji. 2019. Extracting entities and events as a single task using a transition-based neural model. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI-19*, pages 5422–5428. International Joint Conferences on Artificial Intelligence Organization.

Mi Zhang and Tieyun Qian. 2020. Convolution over hierarchical syntactic and lexical graphs for aspect level sentiment analysis. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 3540–3549, Online. Association for Computational Linguistics.

Wenxuan Zhang, Yang Deng, Xin Li, Yifei Yuan, Lidong Bing, and Wai Lam. 2021a. Aspect sentiment quad prediction as paraphrase generation. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 9209–9219, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Wenxuan Zhang, Xin Li, Yang Deng, Lidong Bing, and Wai Lam. 2021b. Towards generative aspect-based sentiment analysis. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 504–510, Online. Association for Computational Linguistics.

Zheng Zhang, Zili Zhou, and Yanna Wang. 2022. SSEGCN: Syntactic and semantic enhanced graph convolutional network for aspect-based sentiment analysis. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4916–4925, Seattle, United States. Association for Computational Linguistics.

A Tree Pruning

As the original opinion trees are too complicated for parsing, we adopt a pruning method to reduce the duplication in trees. To be more specific, we introduce our method with a pruning example of review "So *happy* to have a *great* bar", which can be described as following steps, and the original tree is demonstrated in Figure 7(a).

- The unary chain of category and polarity are integrated into the aspect node and opinion node respectively. The processed result is shown in Figure 7(b).
- We delete the chains with ϵ leaf node, the processed result is shown in Figure 7(c).
- If the children nodes contain nodes that have exactly the same node type with the parent node, we will delete the parent node and connect children with the ancestor node directly, the processed result is shown in Figure 7(d).

Therefore, Figure 7(d) gives the final formation of our opinion tree for parsing.

B Case Study

We launch a set of case studies to make a more intuitive comparison between our model and OTG (Bao et al., 2022). We select reviews that are predicted into invalid formation by OTG to demonstrate our models' superiority in guaranteeing structure well-formedness. As demonstrated in Table 8, these cases can be divided into following categories:

Invalid Term

The first three examples are about invalid terms which generated from OTG.

In the first example, OTG gives a very typical wrong prediction, it rewrites "*waiting*" to "*wait*", which could change the original meanings and does not meet the requirement of extracting raw text from the review, while our method operating over raw spans, easily gives a right answer.

In the second example, OTG generates "*atmosphere*" as the aspect term based on its understanding of "*feeling*" since they have similar semantic information. However, '*atmosphere*" does not exist in the review. On the other hand, our model also shots the right target but selects it as the final prediction under the constraints of chart decoder.

In the third example, OTG generates "*not that slow*" from the review, which are not continuous in the original text: the words "*not that*" appear in the beginning but "*slow*" appears in the end. In this situation, our span-based method can easily extract "*slow*" as the opinion term since it can only operate over raw spans.

Invalid Structure

The invalid structure means that the output sequence of OTG can not be recovered into a valid tree structure, this may due to various reasons. One of the common reasons is unmatched brackets. The fourth example shows an OTG's output sequence that can not be decoded into a valid tree since the sequence that starts with "opinion" can not be recognized as a subtree. In contrast, with the CYK-style algorithm, our method build trees and subtrees over spans, ensuring the legality of trees or subtrees.

Invalid Category

OTG also would classifies aspect term into a non-existing category. In the fifth example, the aspect term "*msi headset*" is classified into a non-existing category "HEADSET GENERAL" by OTG, which usually happens when it comes to the generative method with LAPTOP dataset since it has more than 100 categories. This would not be a difficult problem for our model's classifier, it will set specific target classes before starting the training process.

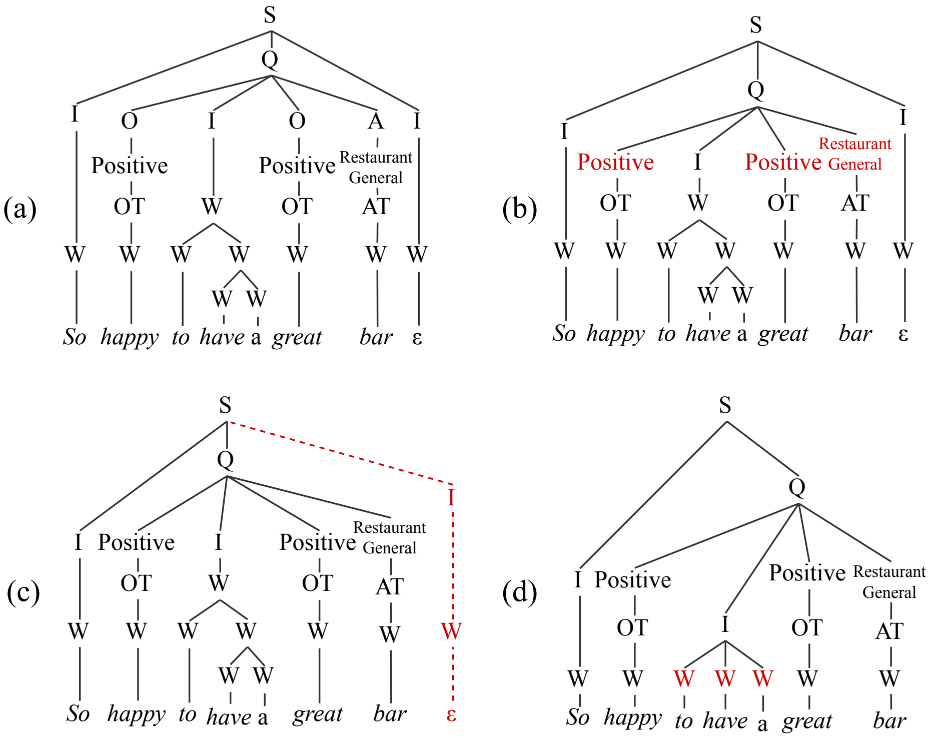

Figure 7: Example of tree pruning.

Review text	**Reason**	**OTG**	**Ours**
The waiting staff has been perfect	Invalid Term	SERVICE GENERAL✓ wait staff ✗ POSITIVE ✓ perfect ✓	SERVICE GENERAL ✓ waiting staff ✓ POSITIVE ✓ perfect ✓
I also really enjoy the intimate feeling of a small restaurant.	Invalid Term	AMBIENCE GENERAL✓ atmosphere✗ POSITIVE ✓ intimate ✓	AMBIENCE GENERAL ✓ feeling ✓ POSITIVE ✓ intimate ✓
not that this machine boots up slow.	Invalid Term	OS PERFORMANCE✗ boots up✓ neural ✗ not that slow ✗	LAPTOP PERFORMANCE ✓ boots up ✓ NEGATIVE ✓ slow ✓
we're can't say enough about their delicious gourmet pizza ' s!	Invalid structure	(root (quad (aspect (food quality, pizza))), (opinion (positive, null))))✗	FOOD QUALITY✓ delicious ✓ POSITIVE ✓ pizza ' s ✓
writing this review so early to receive that nice msi headset.	Invalid category	HEADSET GENERAL✗ msi headset✓ POSITIVE ✓ nice ✓	DEVICE GENERAL ✓ msi headset✓ POSITIVE ✓ nice ✓

Table 8: Case study

From the cases shown in Table 8, we can find that our method shows significant superiority in modeling tree structural constraints and guaranteeing tree structure well-formedness, along with the quality of recovery from tree structure to sentiment elements, while OTG has to employ complex post-processing method to strengthen its shortage.

ACL 2023 Responsible NLP Checklist

A For every submission:

☑ A1. Did you describe the limitations of your work?
Section 9.

☑ A2. Did you discuss any potential risks of your work?
Section 9.

☑ A3. Do the abstract and introduction summarize the paper's main claims?
Abstract and Section 1.

☒ A4. Have you used AI writing assistants when working on this paper?
N/A.

B ☑ Did you use or create scientific artifacts?

Section 3 and 4.

☑ B1. Did you cite the creators of artifacts you used?
Section 6.

☑ B2. Did you discuss the license or terms for use and / or distribution of any artifacts?
Section 6.

☑ B3. Did you discuss if your use of existing artifact(s) was consistent with their intended use, provided that it was specified? For the artifacts you create, do you specify intended use and whether that is compatible with the original access conditions (in particular, derivatives of data accessed for research purposes should not be used outside of research contexts)?
Section 6.

☑ B4. Did you discuss the steps taken to check whether the data that was collected / used contains any information that names or uniquely identifies individual people or offensive content, and the steps taken to protect / anonymize it?
Section 6.

☑ B5. Did you provide documentation of the artifacts, e.g., coverage of domains, languages, and linguistic phenomena, demographic groups represented, etc.?
Section 6.

☑ B6. Did you report relevant statistics like the number of examples, details of train / test / dev splits, etc. for the data that you used / created? Even for commonly-used benchmark datasets, include the number of examples in train / validation / test splits, as these provide necessary context for a reader to understand experimental results. For example, small differences in accuracy on large test sets may be significant, while on small test sets they may not be.
Section 6.

C ☑ Did you run computational experiments?

Section 6.

☑ C1. Did you report the number of parameters in the models used, the total computational budget (e.g., GPU hours), and computing infrastructure used?
Section 6.

The Responsible NLP Checklist used at ACL 2023 is adopted from NAACL 2022, with the addition of a question on AI writing assistance.

☑ C2. Did you discuss the experimental setup, including hyperparameter search and best-found hyperparameter values?
Section 6.

☑ C3. Did you report descriptive statistics about your results (e.g., error bars around results, summary statistics from sets of experiments), and is it transparent whether you are reporting the max, mean, etc. or just a single run?
Section 6.

☑ C4. If you used existing packages (e.g., for preprocessing, for normalization, or for evaluation), did you report the implementation, model, and parameter settings used (e.g., NLTK, Spacy, ROUGE, etc.)?
Section 6.

D ☒ Did you use human annotators (e.g., crowdworkers) or research with human participants?

Left blank.

□ D1. Did you report the full text of instructions given to participants, including e.g., screenshots, disclaimers of any risks to participants or annotators, etc.?
No response.

□ D2. Did you report information about how you recruited (e.g., crowdsourcing platform, students) and paid participants, and discuss if such payment is adequate given the participants' demographic (e.g., country of residence)?
No response.

□ D3. Did you discuss whether and how consent was obtained from people whose data you're using/curating? For example, if you collected data via crowdsourcing, did your instructions to crowdworkers explain how the data would be used?
No response.

□ D4. Was the data collection protocol approved (or determined exempt) by an ethics review board?
No response.

□ D5. Did you report the basic demographic and geographic characteristics of the annotator population that is the source of the data?
No response.