Text Generation Model Enhanced with Semantic Information in Aspect Category Sentiment Analysis

Tu Dinh Tran   Kiyoaki Shirai   Natthawut Kertkeidkachorn
Japan Advanced Institute of Science and Technology
1-1 Asahidai, Nomi, Ishikawa 923-1292, Japan
{s2110422, kshirai, natt}@jaist.ac.jp

Abstract

Aspect Category Sentiment Analysis (ACSA) is one of the main subtasks of sentiment analysis, which aims at predicting polarity over a given aspect category. Recently, generative methods have emerged as an efficient way to use a pre-trained language model for ACSA. However, those methods fail to model relations between target words and opinion words in a sentence including multiple aspects. To tackle this problem, this paper proposes a method to incorporate Abstract Meaning Representation (AMR), which describes the semantic representation of a sentence as a directed graph, into a text generation model. Furthermore, two regularizers are designed to guide the allocation of cross attention weights over AMR graphs. One is the identical regularizer, which constrains the attention weights of aligned nodes, the other is the entropy regularizer, which helps the decoder generate tokens by only giving a high degree of consideration to a few related nodes in the AMR graph. Experimental results on three datasets show that the proposed method outperforms state-of-the-art methods, proving the effectiveness of our model.

1 Introduction

Aspect based sentiment analysis is an important task, which analyzes sentiments regarding an aspect of a product or a service. This task includes many subtasks, such as Aspect Category Detection (ACD) and Aspect Category Sentiment Analysis (ACSA). ACD is the task of detecting aspect categories while ACSA concentrates on predicting the polarity of given aspect categories. This study focuses on ACSA only. Figure 1 shows an example of ACSA task, where *negative* and *positive* are the polarities of the two provided categories *service* and *food*.

The conventional approaches carry out ACSA as a classification task. Wang et al. (2016) and Cheng et al. (2017) use an attention mechanism for discovering aspect-related words. To achieve better representations, pre-trained language models such as BERT (Devlin et al., 2019) are used (Sun et al., 2019; Jiang et al., 2019). Although achieving competitive results, the fine-tuning of a pre-trained language model for ACSA suffers from two drawbacks: the difference between fine-tuning and pre-training tasks and the gap between newly initialized classification layers and the pre-trained model. Such inconsistency is often harmful to the training of an outstanding classifier for ACSA.

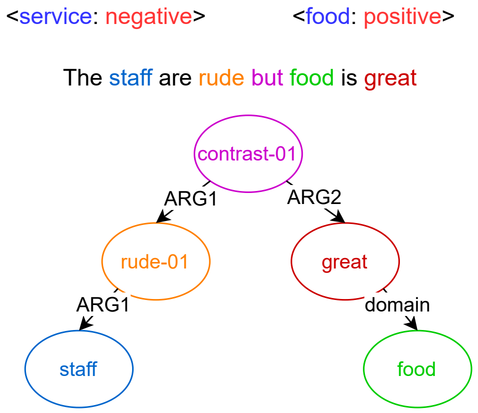


Figure 1: Example of ACSA with the corresponding AMR graph and alignment with the review sentence.

To solve the above problems, Liu et al. (2021) propose to transform the sentiment classification task into text generation, which better leverages the power of pre-trained language models following the seq2seq framework like BART (Lewis et al., 2020). However, the naive text generation method cannot fully capture relations between opinion words and target words in sentences containing multiple aspects.

Abstract Meaning Representation (AMR, Banarescu et al., 2013), which is a semantic representation of a sentence in the form of rooted, labeled, directed and acyclic graphs, can model the relations between the target words and associated opinion words. For example, in Figure 1, the relation be-

Findings of the Association for Computational Linguistics: ACL 2023, pages 5256–5268
July 9-14, 2023 ©2023 Association for Computational Linguistics

tween "*staff*" and "*rude*" is captured by a directed edge between two nodes "*staff*" and "*rude-01*". In addition, the AMR graph also provides the high level semantic information, which means words with the same meaning, like "*but*", "*although*" and "*nevertheless*", could be represented by the same node "*contrast-01"*. This paper investigates the potential of combining AMR with the naive text generation model to perform ACSA.

Furthermore, we also design two regularizers for guiding cross attentions over the AMR graph of the decoder. We observe that words (in a sentence) and nodes (in an AMR graph) that are semantically similar should be paid the same amounts of attention. Therefore, we minimize the difference of two cross attentions over aligned words and AMR nodes. Moreover, the decoded tokens should only be attentive to related AMR nodes. To achieve that, we minimize the entropy of the cross attentions over the AMR graph in the decoder layers.

We evaluate our model using the Rest14, Rest14-hard and MAMS benchmark datasets. The results show that our model is better than the baselines and achieves a state-of-the-art performance.

Our contributions can be summarized as follows:

- We propose a model that incorporates an AMR graph encoder within a seq2seq framework for capturing the relations between the target words and opinion words and using this semantic information for ACSA. To the best of our knowledge, this is the first attempt to explore how to use AMR in ACSA.

- We propose two regularizers to improve the cross attention mechanism over the AMR graph using AMR alignments and information entropy.

- We demonstrate the effectiveness of our proposed method through experiments on three datasets.

2 Related Work

Aspect Category Sentiment Analysis Numerous attempts have been made to improve ACSA. Wang et al. (2016) propose an LSTM-based model combined with an attention mechanism to attend for suitable words with given aspects. Ruder et al. (2016) capture inter-sentence relations within a review using a hierarchical bidirectional LSTM model. Xue and Li (2018) extract features using CNN and output features related to the categories of aspects by using a gated mechanism. Xing et al. (2019), Liang et al. (2019) and Zhu et al. (2019) incorporate aspect category information into a sentence decoder for generating representations specific to both the aspect and its context. Sun et al. (2019) construct auxiliary sentences from aspects for performing ACSA as a sentence-pair classification task. Jiang et al. (2019) propose a new capsule network which captures relationship between the aspects and the contexts. Li et al. (2020b) aggregate the sentiments of the words that indicate an aspect category, so as to predict the sentiment of this category. Liu et al. (2021) use a template-based method to perform ACSA as a generation task; this can leverage the knowledge of a pre-trained language model. Shan et al. (2022) use additional syntactic information to enhance the sentiment features. Liu et al. (2023) utilize commonsense knowledge graph and data augmentation to overcome the shortage of training data.

To avoid error propagation, joint models which perform ACSA and ACD simultaneously have been proposed. Schmitt et al. (2018) propose two models with LSTM and CNN, which output an aspect category and its corresponding polarity at the same time. Hu et al. (2019) apply orthogonal and sparseness constraints on attention weights. Wang et al. (2019) design an AS-Capsules model to explore the correlation of aspects with sentiments through share modules. Li et al. (2020a) propose a joint model with a shared sentiment prediction layer.

AMR With the development of AMR parsers, AMR-to-text generation models and larger parallel datasets, AMR has been applied successfully to many downstream text generation tasks. For example, it has been integrated into a machine translation model as additional information for the source side (Song et al., 2019; Nguyen et al., 2021; Xu et al., 2021). In text summarization, several researchers transform AMR representations of sentences into an AMR graph of a summary and generate a text summary from the extracted subgraph (Liu et al., 2015; Dohare and Karnick, 2017; Hardy and Vlachos, 2018; Inácio and Pardo, 2021). However, as explained in Section 1, there has been no attempt to apply AMR to ACSA.

3 Text Generation Model for ACSA

This section presents an overview of the text generation model for ACSA proposed by Liu et al.

(2021), since our proposed model is based on it. The model follows the seq2seq framework that converts a review sentence to a target sentence indicating the polarity of the aspect. It fine-tunes the pre-trained BART model for the text generation task. The model takes a review sentence $\mathbf{X} = \{x_1, x_2, ..., x_{|X|}\} = x_{1:|X|}$ as an input and generates a target sentence $\mathbf{Y} = \{y_1, y_2, ..., y_{|Y|}\} = y_{1:|Y|}$, where $|X|$ and $|Y|$ are the number of tokens in the source and target sentence respectively.

3.1 Target Sentence Creation

The target sentence is formed by filling an aspect category and a sentiment word into a predefined template. We denote the set of aspect categories by $\mathbf{A} = \{a_1, a_2, ..., a_{|A|}\}$ and the set of sentiment words by $\mathbf{S} = \{s_1, s_2, ..., s_{|S|}\}$. The template is defined manually like "*The sentiment polarity of [ASPECT_CATEGORY] is [SENTIMENT_WORD]*". For each review sentence $\mathbf{X}$ whose corresponding aspect category is a_p and sentiment polarity is s_t, we fill the slots in the template and get the target sentence "*The sentiment polarity of* $\langle a_p \rangle$ *is* $\langle s_t \rangle$" (E.g., "*The sentiment polarity of service is negative*").

3.2 Training and Inference

For the training, given a pair of sentences $(\mathbf{X}, \mathbf{Y})$, the method fetches the input sentence $\mathbf{X}$ into the encoder to get vector presentation h^{enc} of $\mathbf{X}$ as in Equation (1). In the decoder, the hidden vector at a time step j is calculated using h^{enc} and the hidden vectors of the previous time steps, as in Equation (2).

$$h^{enc} = \text{Encoder}(x_{1:|X|}) \tag{1}$$

$$h_j^{dec} = \text{Decoder}(h^{enc}, h_{1:j-1}^{dec}) \tag{2}$$

The conditional probability of the output token y_j is:

$$P(y_j|y_{1:j-1}, x_{1:|X|}) = \text{softmax}(\mathbf{W}h_j^{dec} + \mathbf{b}), \tag{3}$$

where $\mathbf{W} \in \mathbb{R}^{d_h \times |\mathcal{V}|}$ and $\mathbf{b} \in \mathbb{R}^{|\mathcal{V}|}$, $|\mathcal{V}|$ represents the vocabulary size. The loss function of this model is the following Cross Entropy:

$$\mathcal{L}_{ce} = -\sum_{j=1}^{|Y|} \log P(y_j|y_{1:j-1}, x_{1:|X|}). \tag{4}$$

For inference, we calculate the probabilities of all possible target sentences with different sentiment polarity classes using the trained model and choose the one with the highest probability. For an input sentence $\mathbf{X}$, aspect category a_p and sentiment polarity s_t, the probability of a target sentence $\mathbf{Y}_{a_p,s_t} = \{y_1, y_2, ..., y_m\}$ is calculated as follows:

$$f(\mathbf{Y}_{a_p,s_t}) = \sum_{j=1}^{m} \log P(y_j|y_{1:j-1}, \mathbf{X}) \tag{5}$$

4 Proposed Method

Figure 2 shows our proposed model, which follows general text generation methods (Liu et al., 2021). To encode semantic information from an AMR graph, we use a graph encoder module (Subsection 4.1). We incorporate that information by adding a new cross attention layer to the decoder (Subsection 4.2). We also introduce two types of regularizers to guide the attention score of the new cross attention layer (Subsection 4.3). In addition, Subsection 4.4 introduces the loss function of the model, and 4.5 presents the pre-training procedure to overcome the difficulty of training newly initialized layers and a pre-trained language model.

4.1 AMR Encoder

The AMR encoder adopts Graph Attention Networks (GAT, Velickovic et al., 2018). For a given input sequence $\mathbf{X} = \{x_1, x_2, ...x_{|X|}\}$, we construct a corresponding AMR graph $\mathbf{G} = (\mathbf{V}, \mathbf{E})$ from the pre-trained AMR parser (Bai et al., 2022), where $\mathbf{V} = \{v_1, v_2, ..., v_{|V|}\}$ is the set of nodes and $\mathbf{E} \in \mathbb{R}^{|V| \times |V|}$ is the adjacency matrix presenting the relations between the nodes. We treat the AMR graph as an undirected graph, which means $e_{ij} = e_{ji} = 1$ if the two nodes v_i and v_j are connected, otherwise 0.

Given a graph $\mathbf{G} = (\mathbf{V}, \mathbf{E})$ and node $v_i \in \mathbf{V}$, we can obtain h'_i, the hidden state of node v_i, as follows:

$$h'_i = \sigma\Big(\sum_{j \in \mathcal{N}_i} \alpha_{ij} \mathbf{W} h_j\Big) \tag{6}$$

$$\alpha_{ij} = \frac{\exp\big(\sigma\big(\mathbf{a}^T[\mathbf{W}h_i \| \mathbf{W}h_j]\big)\big)}{\sum_{k \in \mathcal{N}_i} \exp\big(\sigma\big(\mathbf{a}^T[\mathbf{W}h_i \| \mathbf{W}h_k]\big)\big)}, \tag{7}$$

where $\mathbf{a}^T$ and $\mathbf{W}$ are trainable parameters, σ is the LeakyRELU function, $\|$ denotes the concatenation of two vectors, $\mathcal{N}_i$ is the set of neighbor nodes of v_i in $\mathbf{G}$, and h_i is the initial representation of v_i. Note that a node (word) consists of several subwords in general. Using the embedding of the AMR parser, h_i is defined as the average of the subword vectors.

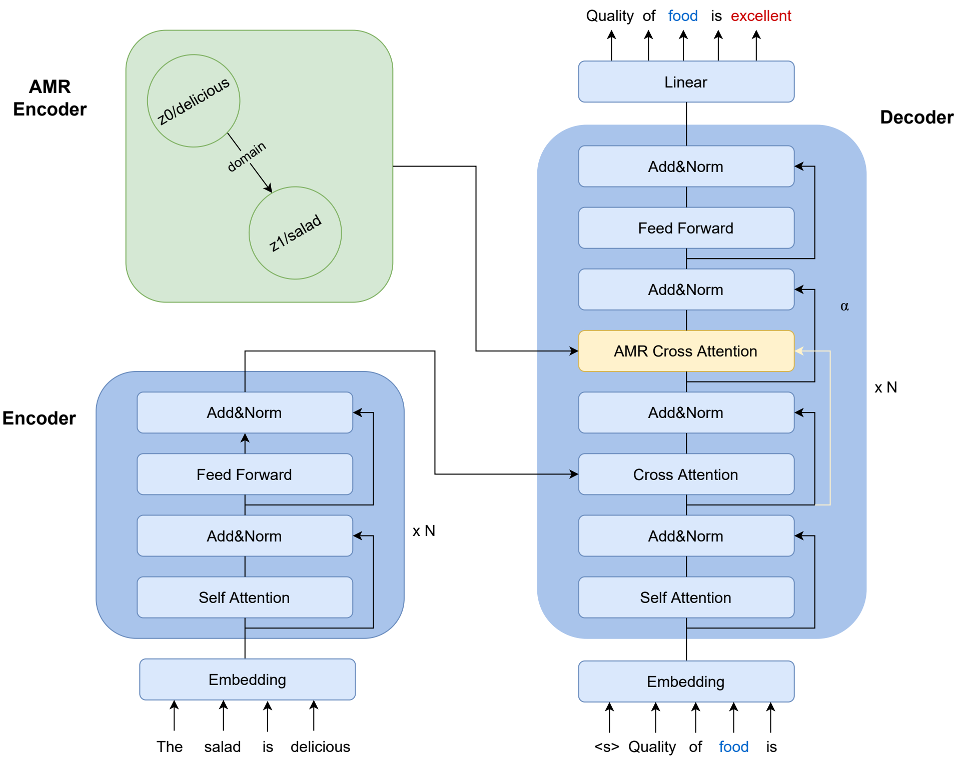


Figure 2: Architecture of our proposed model.

Applying the multi-head attention mechanism from the Transformer architecture (Vaswani et al., 2017), we obtain the updated representation of node v_i :

$$h'_i = \Big\Vert_{k=1}^{K} \sigma\Big(\sum_{j \in \mathcal{N}_i} \alpha_{ij}^k \mathbf{W}^k h_j \Big), \tag{8}$$

, where K is the number of attention heads, α_{ij}^k are the attention coefficients of the k-th head, $\mathbf{W}^k$ is the weight matrix at the k-th head, and $\Vert$ stands for the concatenation of multiple vectors.

4.2 Decoder

After obtaining the graph information, we feed it into each decoder layer by adding a new cross attention module for AMR referred to as "AMR Cross Attention" in Figure 2. We write h' for the representations of the AMR nodes obtained from GAT, x is the vector representation of the input sentence and y^l is the output of l-th decoder layer. The output of the $(l+1)$-th decoder layer, y^{l+1}, is obtained as follows:

$$\dot{y}^l = \text{LN}(y^l + \text{SelfAttn}(y^l)) \tag{9}$$
$$\ddot{y}^l = \text{LN}(\dot{y}^l + \text{CrossAttn}(\dot{y}^l, x)) \tag{10}$$
$$\dddot{y}^l = \text{LN}(\ddot{y}^l + \text{CrossAttn}(\ddot{y}^l, h')) \tag{11}$$
$$y^{l+1} = \text{LN}(\dddot{y}^l + \text{FFN}(\dddot{y}^l)), \tag{12}$$

where LN is the layer normalization function, SelfAttn is the self-attention module, CrossAttn is the cross-attention module, and FFN is the feed-forward neural network.

Training a deep model like Transformer is really hard and even harder with one more cross-attention module. To overcome this difficulty, we employ ReZero (Bachlechner et al., 2021) as the AMR cross attention module instead of the normal residual module. This method is implemented as follows:

$$\tilde{y}^l = \ddot{y}^l + \alpha \mathcal{F}(\ddot{y}^l), \tag{13}$$

where $\mathcal{F}$ denotes non-trivial functions and α is a trainable parameter which helps moderate the updating of the AMR cross attention.

4.3 AMR Cross Attention Regularizers

To incorporate the semantic information from the AMR graph more effectively, we propose two regularizers over the attention scores of the AMR cross attention module.

Identical Regularizer Intuitively, a word in a sentence and its aligned node in the AMR graph should receive the same attention as they are supposed to represent similar semantic information. Two transformation matrices for the cross attention matrix over each of the source (input) sentences and the AMR graphs, $\text{align}_{src} \in R^{|X|\times|P|}$ and $\text{align}_{amr} \in R^{|V|\times|P|}$, respectively, are defined in Equations (14) and (15), where $|P|$ is the number of aligned pairs of words and nodes.

$$align_{src}[i,k] = \begin{cases} \frac{1}{|\mathcal{T}_i|} & \text{if token } x_i \text{ belongs to an} \\ & \text{aligned word at position } k \\ 0 & \text{otherwise} \end{cases} \tag{14}$$

$$align_{amr}[j,k] = \begin{cases} 1 & \text{if node } v_j \text{ is aligned} \\ & \text{at position } k \\ 0 & \text{otherwise} \end{cases} \tag{15}$$

Here, $\mathcal{T}_i$ denotes a set of subwords in the aligned word. With these matrices and two given cross attention matrices $A_{src} \in \mathbb{R}^{|Y|\times|X|}$, $A_{i\_amr} \in \mathbb{R}^{|Y|\times|V|}$ over the review sentence and the AMR graph, respectively, the identical regularizer is formulated as follows:

$$\mathcal{L}_{ir} = \sum_{i=1}^{L} \frac{1}{L} \| A^i_{src} \cdot align_{src} - A^i_{i\_amr} \cdot align_{amr} \|_F, \tag{16}$$

where $\|\|_F$ denotes the Frobenius norm and L is the number of the decoder layers. The matrix A_{src} is obtained from an oracle fine-tuned text generation model. Also, the matrix A_{i_amr} is obtained by fetching the same input with the regular cross attention layer over the source sentence, which is indicated by the yellow line in Figure 2.

Entropy Regularizer We expect that our model concentrates on a few important nodes. This means that the cross attention distribution of the tokens over the AMR nodes is supposed to be skewed. Therefore, we try to minimize the information entropy (Shannon, 1948) of the attention scores of the tokens over the AMR nodes. We first calculate the mean of the cross attention score of the token i at the node j over $\mathcal{H}$ attention heads as follows:

$$\tilde{a}_{ij} = \frac{1}{\mathcal{H}} \sum_{h=1}^{\mathcal{H}} a_{ijh} \tag{17}$$

Then, the entropy of the l-th decoder layer is calculated over $|V|$ nodes and $|Y|$ output tokens:

$$H^l = -\frac{1}{|Y|} \sum_{i}^{|Y|} \sum_{j}^{|V|} \tilde{a}_{ij} \log \tilde{a}_{ij} \tag{18}$$

The entropy regularizer is defined as the mean entropy of the L decoder layers:

$$\mathcal{L}_{er} = \frac{1}{L} \sum_{l=1}^{L} H^l \tag{19}$$

4.4 Loss Function

For training the proposed model, the loss function is the sum of the normal cross entropy loss and the aforementioned two regularizers:

$$\mathcal{L} = \mathcal{L}_{ce} + \lambda_1 \mathcal{L}_{ir} + \lambda_2 \mathcal{L}_{er}, \tag{20}$$

where λ_1 is the scaling factors of the identical regularizer and λ_2 is that of the entropy regularizer.

4.5 Pre-training

It is hard to fine-tune our model, which consists of randomly initialized modules like the AMR graph encoder and the AMR cross attention together with the pre-trained BART. Following (Bataa and Wu, 2019) and (Gheini et al., 2021), which show the positive effect of pre-training a language model with in-domain data and fine-tuning a cross attention layer, after initializing the whole model, we train it with text denoising tasks using review sentences. Following BART, we add noise into the input sentences using the following three methods:

- **Token Masking**: random tokens are sampled and replaced by $[MASK]$ token.
- **Word Deletion**: instead of deleting subwords like BART, the whole of text spans is deleted in this method.
- **Text Infilling**: random text spans are replaced by $[MASK]$ using a Poisson distribution.

Algorithm 1 shows the pseudocode of the text corruption algorithm that adds noise by the above three methods.

Algorithm 1 Text corruption algorithm

```
Input: review sentence $\mathbf{X} = \{x_1, x_2, ...x_n\}$
Output: corrupted review sentence $\mathbf{X'} = \{x'_1, x'_2, ...x'_m\}$
$p_{token} \leftarrow 0.15$ - probability of replacing one token by [MASK]
$p_{word} \leftarrow 0.3$ - probability of replacing text spans by [MASK]
$\lambda_{Possion} \leftarrow 3$ - value for $\lambda$ parameter in Possion distribution
$p \leftarrow gen\_random[0, 1]$
if $p < \frac{1}{3}$ then
    $\mathbf{X'} \leftarrow mask\_tokens(\mathbf{X}, p_{token})$
else if $p < \frac{2}{3}$ then
    $\mathbf{X'} \leftarrow mask\_text\_spans(\mathbf{X}, p_{word}, \lambda_{Possion})$ - Sampling text spans with span lengths drawn from a Poisson distribution and masking them.
else
    $\mathbf{X'} \leftarrow mask\_text\_spans(X, p_{word}, \lambda_{Possion})$
    $\mathbf{X'} \leftarrow delete\_mask\_tokens(\mathbf{X'})$
end if
```

The entropy regularizer is also taken into account in this pre-training step. That is, the loss function of the pre-training is defined as:

$$\mathcal{L} = \mathcal{L}_{ce} + \lambda_3 \mathcal{L}_{er}, \tag{21}$$

where λ_3 is the scaling factor for the entropy regularizer.

5 Experiments

5.1 Dataset

The following three datasets are used in our experiments.

- **Rest14**: This dataset consists of reviews in the restaurant domain, which is included in the Semeval-2014 task (Pontiki et al., 2014). Samples labeled with "conflict" are removed, so the remaining samples have the labels "positive", "negative" and "neutral". In addition, we follow the splitting of the development set suggested by Tay et al. (2018) for the sake of a fair comparison.

- **Rest14-hard**: Xue and Li (2018) construct this dataset for better evaluating a model on sentences with multiple aspects. The training set and development set are the same as those of Rest14.

- **MAMS-ACSA**: For the same purpose as the Rest14-hard, Jiang et al. (2019) propose a larger dataset for ACSA in which each sentence contains at least two different aspects.

Their details are shown in Table 1.

Dataset		**#Pos**	**#Neg**	**#Neu**
Rest14	Train	1855	733	430
	Dev	324	106	70
	Test	657	222	94
Rest14-hard	Test	21	20	12
MAMS-ACSA	Train	1929	2084	3077
	Dev	241	259	388
	Test	245	363	393

Table 1: Statistics of datasets.

5.2 Baselines

We compare our method with multiple baselines:

- **GCAE** (Xue and Li, 2018): employs CNN model with gating mechanism to selectively output the sentiment polarity related to a given aspect.

- **AS-Capsules** (Wang et al., 2019): exploits the correlation between aspects and corresponding sentiments through a capsule-based model.

- **CapsNet** (Jiang et al., 2019): is the capsule network based model to learn the relations between the aspects and the contexts.

- **CapsNet-BERT** (Jiang et al., 2019): is the CapsNet model based on the pre-trained BERT.

- **BERT-pair-QA-B** (Sun et al., 2019): performs ACSA as the sentence pair classification task by fine-tuning of the pre-trained BERT.

- **AC-MIMLLN** (Li et al., 2020b): predicts the polarity of a given aspect by combining the sentiments of the words indicating the aspect.

- **AC-MIMLLN-BERT** (Li et al., 2020b): is the AC-MIMLLN model based on the pre-trained BERT.

- **BART generation** (Liu et al., 2021): performs ACSA by a text generation model with the pre-trained BART. It is almost equivalent to our model without AMR.

- **BART generation with pre-training**: is the BART generation model combined with our pre-training method except for applying entropy regularization.

5.3 Implementation Details

The template used for constructing the target sentences in our experiments is "*Quality of [ASPECT_CATEGORY] is [SENTIMENT_WORD]*". The *[ASPECT_CATEGORY]* is filled by the aspect word, while the *[SENTIMENT_WORD]* is filled by one of $\{excellent, awful, fine\}$ which corresponds to $\{positive, negative, neutral\}$ respectively. For AMR parsing, we use the pre-trained model of AMRBART[1] (Bai et al., 2022). In addition, LEAMR[2] (Blodgett and Schneider, 2021) is adopted to align the words in the input sentence and the nodes in the AMR graph.

In the pre-training step, we initialize the parameters of BART using the checkpoint of BART base[3]. Unlike the parameters of BART, the parameters of the AMR graph encoder and the AMR cross attention modules are newly initialized with the uniform distribution. After pre-training, the last checkpoint is used for fine-tuning the ACSA model. The Adam optimizer (Kingma and Ba, 2015) is used for optimizing the model. The original parameters of BART's encoder and decoder are trained with a learning rate 2e-5 while the learning rate is set to 3e-5 for the parameters in the AMR graph encoder and the AMR cross attention modules. We set the number of the attention heads of AMR encoder to 6, the number of AMR cross attention heads to 6, the batch size is 16 and the dropout value 0.1. The initial value for the ReZero weight α is 1. The regularization coefficients λ_1 and λ_2 are set to (0.075, 0.1), (0.075, 0.1) and (0.025, 0.0075) for the three datasets, while λ_3 is always set to 5e-3. All hyperparameters are tuned based on the accuracy on the development set.

5.4 Experimental Results

The results of the experiments are presented in Table 2. The models were trained and evaluated five times with different initializations of the parameters. The table shows the average and standard deviation of the accuracy of five trials using the format "mean (±std)". First, our model outperforms all baselines on the three datasets, which indicates the necessity of incorporating the semantic information into the text generation model for ACSA. Second, compared with the models that learn relations between the aspect and the context like CapsNet, AC-MIMLLN, BERT-pair-QA-B and BART generation, the dominance of our model proves that exploiting the AMR graph to learn relations between words is a better way to capture contextual information. The fact that our model also outperforms BART generation with the pre-training further supports the effectiveness of the AMR. Third, the competitive results over the Rest14-hard and MAMS datasets show the effectiveness of the identical and entropy regularizers in enabling the model to concentrate on the correct aspect-related nodes, which is essential for identification of the polarity over multiple aspects.

5.5 Ablation Study

To further investigate the effects of the different modules in our model, we conducted ablation studies. The results are presented in Table 3. First, it is found that the removal of the identical regularizer downgrades the performance, which indicates the importance of precisely capturing the semantic information. Second, we also notice that the models without the entropy regularizer perform poorly with reduction by 0.8, 1.1 and 0.4 percentage points in the accuracy on Rest14, Rest14-hard and MAMS, respectively. This shows that the entropy regularizer is essential to prevent models from attending to unnecessary AMR nodes. In addition, removing both regularizers degrades the performance more than removing each of the regularizer, which confirms the essential roles of these regulairzers in performing ACSA. Third, removing the pre-training procedure hurts the performance badly, which leads to decreases by 1.4, 7.5 and 1.5 percentage points on the three datasets respectively. This indicates the big gap between the newly initialized modules and the pre-trained model and the necessity of the pre-training step for overcoming this problem. In summary, the ablation studies show that each component contributes to the entire model. The contribution of the pre-training step is the greatest, while those of the identical and entropy regularizers are comparable to each other.

6 Analysis

6.1 Case Study

To further examine how the semantic information of AMR and two regularizers work well in ACSA, a few examples are shown as a case study. Table 4 compares our model with the state-of-the-art method "BART generation". The symbols P, N

[1] `https://github.com/goodbai-nlp/AMRBART`

[2] `https://github.com/ablodge/leamr`

[3] `https://huggingface.co/facebook/bart-base`

Model	Rest14	Rest14-hard	MAMS
GCAE (Xue and Li, 2018)	81.3(±0.883)	54.7(±4.92)	72.1†
AS-Capsules (Wang et al., 2019)	82.2(±0.414)	60.8(±2.77)	75.1(±0.473)
CapsNet (Jiang et al., 2019)	81.2(±0.631)	54.0(±0.924)	74.0†
AC-MIMLLN (Li et al., 2020b)	81.6(±0.715)	65.3(±2.26)	76.4(±0.704)
BERT-pair-QA-B (Sun et al., 2019)	87.5(±1.18)	69.4(±4.37)	79.1(±0.973)
CapsNet-BERT (Jiang et al., 2019)	86.6(±0.943)	51.3(±1.41)	79.5†
AC-MIMLLN-BERT (Li et al., 2020b)	89.3(±0.720)	74.7(±3.29)	81.2(±0.606)
BART generation (Liu et al., 2021)	90.5(±0.315)	77.4(±2.16)	83.1(±0.478)
BART generation with pre-training	90.6(±0.517)	75.5(±3.77)	83.6(±0.847)
Our model	**91.2(±0.258)**	**78.1(±2.53)**	**84.6(±0.453)**

Table 2: Accuracy (%) of ACSA models. † refers to citation from Jiang et al. (2019).

Model	Rest14	Rest14-hard	MAMS
Our model	**91.2(±0.258)**	**78.1(±2.53)**	**84.6(±0.453)**
w/o identical regularizer	91.0(±0.424)	77.4(±1.89)	84.0(±0.320)
w/o entropy regularizer	90.4(±0.162)	77.0(±1.68)	84.2(±1.10)
w/o entropy and identical regularizer	90.3(±0.426)	74.3(±1.69)	83.8(±0.638)
w/o pre-training	89.8(±0.217)	70.6 (±1.03)	83.1(±0.618)

Table 3: Ablation study.

and O represent the positive, negative and neutral class respectively. The first example, "*I never had an orange donut before so I gave it a shot*", has no explicit sentiment expression. With the help of semantic information and two regularizers, our model can correctly predict the true label while BART generation cannot. The second and third examples contain multiple aspects, which can affect each other's predictions. In the second example, the BART generation model may capture the positive sentiment toward the aspect word "*atmosphere*" for anticipating the sentiment of the different aspect "*service*', which leads to outputting the wrong label. Another incorrect prediction by this baseline is shown in the third example, where the polarities of "*food*" and "*staff*" are mistakenly swapped. In contrast, our model pays attention to only the aspect-related AMR nodes, resulting in the correct predictions in both examples. However, our model also faces a difficulty in some cases. In the last example, it wrongly predicts the sentiment polarity for "*miscellaneous*" because it is really hard to capture aspect-related AMR nodes for a coarse-grained aspect class like "*miscellaneous*".

6.2 Attention Visualization

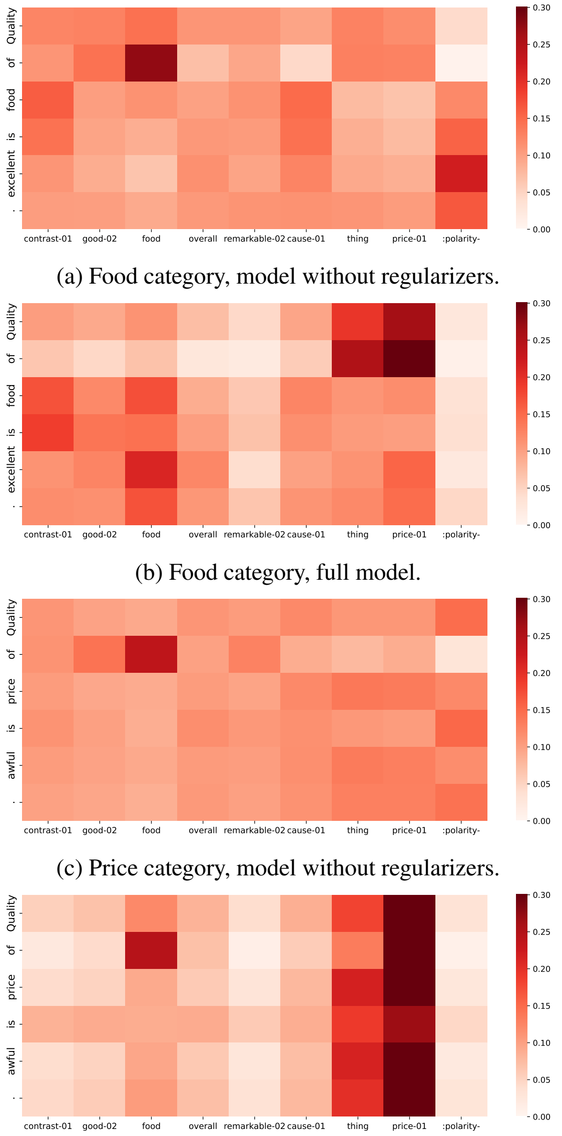
To study the effectiveness of the two regularizers in guiding the AMR cross attention collocation, we illustrate the cross attention matrix produced by our full model and the model without two regularizers in Figure 3. The review sentence is "*The food was good overall, but unremarkable given the price.*". The polarity label of the aspect category "*food*" is positive and the polarity of the aspect category "*price*" is negative. The model without two regularizers has dense attention matrices that might be noise for prediction of the polarity. In contrast, the attention matrices of our full model are sparse. For example, as for the food category, the word "*food*" and "*excellent*" in the target sentence pay much attention or more attention than the model without the regularizers to the nodes "*food*" and "*good-02*" in the AMR graph. Similarly, as for the price category, "*price*" in the target sentence pays a great deal of attention to the node "*price-01*" in the AMR graph, while "*awful*" pays less attention to "*remarkable-02*" than the model without the regularizers. Those cases indicate that our attention mechanism works well even when a review sentence contains multiple aspects.

7 Conclusions

In this paper we proposed a model which integrated the semantic information from the Abstract Meaning Representation (AMR) and the text generation method for the task of Aspect Category Sentiment Analysis (ACSA). Moreover, to more precisely cap-

Sentence	Aspect Category	BART Generation	Our Model	Label
I never had an orange donut before so I gave it a shot.	{food}	(P)	(O)	(O)
The atmosphere was wonderful, however the service and food were not.	{ambiance, service, food}	(P, P, N)	(P, N, N)	(P, N, N)
There are several specials that change daily, which the servers recite from memory.	{food, staff}	(O, P)	(P, O)	(P, O)
The place was busy and had a bohemian feel.	{place, miscellaneous}	(P, P)	(N, P)	(N, O)

Table 4: Case studies of our model compared with state-of-the-art method.



(a) Food category, model without regularizers.

(b) Food category, full model.

(c) Price category, model without regularizers.

(d) Price category, full model.

Figure 3: Attention scores of target sentences over AMR graph in our models.

ture the semantic correlations between the target words and the AMR nodes, we proposed two regularizers: the identical and entropy regularizers, over the AMR cross attention modules. The experimental results on three datasets showed that our model outperformed all baselines.

8 Limitations

Currently, our model only exploits the direct relations between nodes in the AMR graph. In other words, only one-hop neighborhoods can be considered. However, there are a few cases where an opinion word and a related aspect word can be in a k-hop neighborhood. In the future, we will design a model that can capture long distance relations in the AMR graph. Another limitation is that the errors of the pre-trained AMR parsers and AMR alignment models are propagated to the model as a whole. What is required is to improve the performance of those modules.

References

Thomas Bachlechner, Bodhisattwa Prasad Majumder, Henry Mao, Gary Cottrell, and Julian McAuley. 2021. Rezero is all you need: fast convergence at large depth. In *Proceedings of the Thirty-Seventh Conference on Uncertainty in Artificial Intelligence*, volume 161 of *Proceedings of Machine Learning Research*, pages 1352–1361. PMLR.

Xuefeng Bai, Yulong Chen, and Yue Zhang. 2022. Graph pre-training for AMR parsing and generation. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6001–6015, Dublin, Ireland. Association for Computational Linguistics.

Laura Banarescu, Claire Bonial, Shu Cai, Madalina Georgescu, Kira Griffitt, Ulf Hermjakob, Kevin Knight, Philipp Koehn, Martha Palmer, and Nathan Schneider. 2013. Abstract Meaning Representation for sembanking. In *Proceedings of the 7th Linguistic Annotation Workshop and Interoperability with Discourse*, pages 178–186, Sofia, Bulgaria. Association for Computational Linguistics.

Enkhbold Bataa and Joshua Wu. 2019. An investigation of transfer learning-based sentiment analysis in

Japanese. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4652–4657, Florence, Italy. Association for Computational Linguistics.

Austin Blodgett and Nathan Schneider. 2021. Probabilistic, structure-aware algorithms for improved variety, accuracy, and coverage of AMR alignments. 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 3310–3321, Online. Association for Computational Linguistics.

Jiajun Cheng, Shenglin Zhao, Jiani Zhang, Irwin King, Xin Zhang, and Hui Wang. 2017. Aspect-level sentiment classification with heat (hierarchical attention) network. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*, CIKM '17, page 97–106. Association for Computing Machinery.

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.

Shibhansh Dohare and Harish Karnick. 2017. Text summarization using abstract meaning representation. *CoRR*, abs/1706.01678.

Mozhdeh Gheini, Xiang Ren, and Jonathan May. 2021. Cross-attention is all you need: Adapting pretrained Transformers for machine translation. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 1754–1765, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Hardy Hardy and Andreas Vlachos. 2018. Guided neural language generation for abstractive summarization using Abstract Meaning Representation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 768–773, Brussels, Belgium. 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.

Marcio Inácio and Thiago Pardo. 2021. Semantic-based opinion summarization. In *Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP 2021)*, pages 619–628, Held Online. INCOMA Ltd.

Qingnan Jiang, Lei Chen, Ruifeng Xu, Xiang Ao, and Min Yang. 2019. A challenge dataset and effective models for aspect-based sentiment analysis. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 6280–6285, 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*.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online. Association for Computational Linguistics.

Yuncong Li, Zhe Yang, Cunxiang Yin, Xu Pan, Lunan Cui, Qiang Huang, and Ting Wei. 2020a. A joint model for aspect-category sentiment analysis with shared sentiment prediction layer. In *Proceedings of the 19th Chinese National Conference on Computational Linguistics*, pages 1112–1121, Haikou, China. Chinese Information Processing Society of China.

Yuncong Li, Cunxiang Yin, Sheng-hua Zhong, and Xu Pan. 2020b. Multi-instance multi-label learning networks for aspect-category sentiment analysis. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 3550–3560, Online. Association for Computational Linguistics.

Yunlong Liang, Fandong Meng, Jinchao Zhang, Jinan Xu, Yufeng Chen, and Jie Zhou. 2019. A novel aspect-guided deep transition model for aspect based sentiment analysis. In *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 5569–5580, Hong Kong, China. Association for Computational Linguistics.

Bin Liu, Tao Lin, and Ming Li. 2023. Enhancing aspect-category sentiment analysis via syntactic data augmentation and knowledge enhancement. *Knowledge-Based Systems*, 264:110339.

Fei Liu, Jeffrey Flanigan, Sam Thomson, Norman Sadeh, and Noah A. Smith. 2015. Toward abstractive summarization using semantic representations. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1077–1086, Denver, Colorado. 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.

Long H. B. Nguyen, Viet Pham, and Dien Dinh. 2021. Improving neural machine translation with amr semantic graphs. *Mathematical Problems in Engineering*.

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.

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.

Martin Schmitt, Simon Steinheber, Konrad Schreiber, and Benjamin Roth. 2018. Joint aspect and polarity classification for aspect-based sentiment analysis with end-to-end neural networks. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1109–1114, Brussels, Belgium. Association for Computational Linguistics.

Yongxue Shan, Chao Che, Xiaopeng Wei, Xiaodong Wang, Yongjun Zhu, and Bo Jin. 2022. Bi-graph attention network for aspect category sentiment classification. *Knowledge-Based Systems*, 258:109972.

C. E. Shannon. 1948. A mathematical theory of communication. *Bell System Technical Journal*, 27(4):623–656.

Linfeng Song, Daniel Gildea, Yue Zhang, Zhiguo Wang, and Jinsong Su. 2019. Semantic neural machine translation using AMR. *Transactions of the Association for Computational Linguistics*, 7:19–31.

Chi Sun, Luyao Huang, and Xipeng Qiu. 2019. 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, Minneapolis, Minnesota. 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. *Proceedings of the AAAI Conference on Artificial Intelligence*, 32(1).

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Petar Velickovic, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. 2018. Graph attention networks. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net.

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.

Yequan Wang, Aixin Sun, Minlie Huang, and Xiaoyan Zhu. 2019. Aspect-level sentiment analysis using as-capsules. In *The World Wide Web Conference*, WWW '19, page 2033–2044. Association for Computing Machinery.

Bowen Xing, Lejian Liao, Dandan Song, Jingang Wang, Fuzheng Zhang, Zhongyuan Wang, and Heyan Huang. 2019. Earlier attention? aspect-aware lstm for aspect-based sentiment analysis. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI-19*, pages 5313–5319. International Joint Conferences on Artificial Intelligence Organization.

Dongqin Xu, Junhui Li, Muhua Zhu, Min Zhang, and Guodong Zhou. 2021. XLPT-AMR: Cross-lingual pre-training via multi-task learning for zero-shot AMR parsing and text generation. 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 896–907, 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.

Peisong Zhu, Zhuang Chen, Haojie Zheng, and Tieyun Qian. 2019. Aspect aware learning for aspect category sentiment analysis. *ACM Trans. Knowl. Discov. Data*, 13(6).

ACL 2023 Responsible NLP Checklist

A For every submission:

☑ A1. Did you describe the limitations of your work?
Section 8

□ A2. Did you discuss any potential risks of your work?
Not applicable. Left blank.

☑ 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?
Left blank.

B ☑ Did you use or create scientific artifacts?

Section 4

☑ B1. Did you cite the creators of artifacts you used?
Section 4

□ B2. Did you discuss the license or terms for use and / or distribution of any artifacts?
Not applicable. Left blank.

□ 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)?
Not applicable. Left blank.

□ 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?
Not applicable. Left blank.

☑ B5. Did you provide documentation of the artifacts, e.g., coverage of domains, languages, and linguistic phenomena, demographic groups represented, etc.?
Section 5 and 6 (The performance of our proposed model is reported.)

☑ 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.
The statistics of the datasets we used is written in Subsection 5.1.

C ☑ Did you run computational experiments?

Section 5

☑ C1. Did you report the number of parameters in the models used, the total computational budget (e.g., GPU hours), and computing infrastructure used?
Subsection 5.3

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?
Subsection 5.3

☑ 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 5 and 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.)?
Subsection 5.3

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.