A Hierarchical Sequence-to-Set Model with Coverage Mechanism for Aspect Category Sentiment Analysis

Siyu Wang[1†], **Jianhui Jiang**[1†], **Shengran Dai**[1], **Jiangtao Qiu**[2*]
[1] Gusu Laboratory of Materials, Suzhou, China
[2] Southwestern University of Finance and Economics, Chengdu, China
{wangsiyu2022, jiangjianhui2021, daishengran2021}@gusulab.ac.cn
qiujt_t@swufe.edu.cn

Abstract

Aspect category sentiment analysis (ACSA) aims to simultaneously detect aspect categories and their corresponding sentiment polarities (category-sentiment pairs). Some recent studies have used pre-trained generative models to complete ACSA and achieved good results. However, for ACSA, generative models still face three challenges. First, addressing the missing predictions in ACSA is crucial, which involves accurately predicting all category-sentiment pairs within a sentence. Second, category-sentiment pairs are inherently a disordered set. Consequently, the model incurs a penalty even when its predictions are correct, but the predicted order is inconsistent with the ground truths. Third, different aspect categories should focus on relevant sentiment words, and the polarity of the aspect category should be the aggregation of the polarities of these sentiment words. This paper proposes a hierarchical generative model with a coverage mechanism using sequence-to-set learning to tackle all three challenges simultaneously. Our model's superior performance is demonstrated through extensive experiments conducted on several datasets.

Keywords: Aspect Category Sentiment Analysis, Hierarchical Sequence-to-Set Model, Coverage Mechanism

1. Introduction

Sentiment analysis is the process of analyzing people's emotions, attitudes, opinions, and sentiment expressions in textual reviews (Liu, 2012). Aspect-based sentiment analysis (ABSA) (Pontiki et al., 2014) is a fine-grained sentiment analysis task that involves many subtasks, aspect category detection (ACD) and aspect sentiment classification (ASC) are two of them. ACD detects the aspect categories mentioned in a sentence, and ASC predicts the sentiment polarities according to the detected aspect categories (Zhang et al., 2022). For example, in the sentence "`The food is delicious, but the price is a bit expensive.`", the two aspect categories `(food, price)` are detected by ACD, and the sentiment polarities of detected aspect categories `(positive, negative)` can be predicted by ASC. In this paper, we focus on ACSA, which aims to jointly detect the discussed aspect categories (ACD) and their corresponding sentiment polarities (ASC) (Zhang et al., 2022). For the previous example, ACSA models can directly predict two category-sentiment pairs `(food, positive)` and `(service, negative)`.

Previous studies for ACSA can be categorized into two types: the pipeline and the joint approach. The most straightforward way to address ACSA is the pipeline approach (Brun et al., 2016; Lee et al., 2017; Kumar et al., 2016), which first identified the aspect categories contained in the sentence, then predicted the polarity of the detected aspect categories. Obviously, the result of detecting aspect categories has a great impact on the performance of these approaches. Therefore, many recent researchers handled ACSA in a joint way by the classification method. These approaches (Cai et al., 2020; Hu et al., 2019; Gu and Zhang, 2022; Fu et al., 2021; Li et al., 2020c,b; Wang et al., 2019; Li et al., 2020a; Zhou and Law, 2022) employed multi-label classifiers to tackle the task since a sentence contains one or more category-sentiment pairs. They first obtained the contextual representation of sentences through Long Short-term Memory (LSTM) (Hochreiter and Schmidhuber, 1997), Convolutional Neural Networks (CNN) (Kim, 2014), Gated Recurrent Unit (GRU) (Cho et al., 2014), or

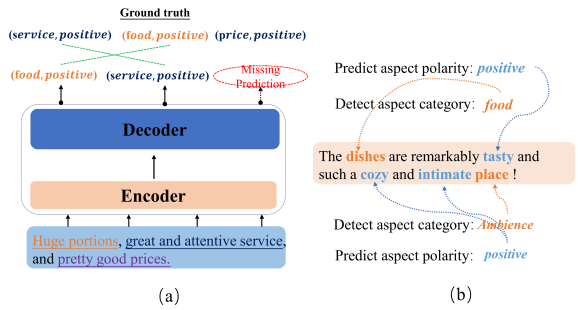

Figure 1: (a) An example of the missing prediction of aspect categories. (b) An example of different aspect categories focus on different sentiment words, where the final polarity is the aggregation of each polarity identified from the sentiment words.

[†]Equal Contribution
[*]Corresponding Author

LREC-COLING 2024, pages 626–635

Bidirectional Encoder Representations from Transformers (BERT) (Devlin et al., 2018). Then these contextual representations were fed into multiple classifiers to identify different aspect categories and corresponding sentiment polarities. In recent years, generative models have achieved very good results in many natural language processing tasks. In view of this, Liu et al. (2021) used the pre-trained generative model BART (Lewis et al., 2020) for ACSA and achieved the best results so far.

However, for ACSA, generative models still face three challenges. **(1)** How to alleviate the missing predictions, namely correctly predicting all category-sentiment pairs contained in a sentence. Take Figure 1(a) as an example, the model detects two pairs but misses the last one. **(2)** Category-sentiment pairs are inherently a disordered set. Consequently, the model incurs a penalty even when its predictions are correct, but the predicted order is inconsistent with the ground truths. For example, as shown in Figure 1(a), the model predicts the sentence contains two category-sentiment pairs `{(service, positive), (food, positive)}`, but the ground truths are `{(food, positive), (service, positive), (price, positive)}`. In this case, although the predictions are correct, the model is still penalized due to the different order of the outputs. **(3)** It is crucial to ensure that different aspect categories focus on different sentiment words, and the polarity of the aspect category should be the aggregation of the polarities of these sentiment words. As shown in Figure 1(b), the sentence "`The dishes are remarkably tasty and such a cozy and intimate place`", for aspect category `food`, its polarity comes from the sentiment word `tasty`. Similarly, the polarity of `ambiance` comes from an aggregation of sentiment words `cozy` and `intimate`.

In this paper, we propose a hierarchical sequence-to-set model with the coverage mechanism (HCSGM) to directly output pairs for addressing the above challenges. HCSGM is based on an autoregressive sequence-to-set architecture. To alleviate the missing predictions, in the decoder, we employ a coverage mechanism (Tu et al., 2016) to memorize the part covered by previous time steps in the sentence. On the other hand, considering the third challenge mentioned before, we design a hierarchical generative mechanism to identify the aspect categories and their sentiment polarities. Finally, a set prediction loss is introduced to optimize the model to avoid the penalty of different prediction orders. Empirical results demonstrate that our models are superior to many state-of-the-art approaches. Our main contributions can be summarized as follows:

- A coverage mechanism is introduced to alleviate the missing predictions when detecting category-sentiment pairs.
- We design a hierarchical generative mechanism to ensure that different aspect categories can focus on relevant sentiment words and aggregate the polarity of the sentiment words.
- A set prediction loss is introduced to train the model and avoid the penalty of different prediction orders.

2. Related Works

In this section, we will introduce a brief review of ACSA. There are two main types of methods for ACSA (Zhang et al., 2022): the pipeline and the joint approach.

2.1. Pipeline Approach

The easiest and most straightforward way to handle ACSA is the pipeline approach (Zhang et al., 2022). For example, XRCE (Brun et al., 2016) and IIT-TUDA (Kumar et al., 2016) heavily depended on feature engineering and divided their pipelines into two separate tasks: aspect detection and aspect polarity classification. Although Lee et al. (2017) incorporated multi-task learning, the prediction of aspect category and polarity still remained separate in the approach. Obviously, the key problem with these approaches is that the performance of aspect detection determines the performance of the entire model. In other words, errors in aspect detection can affect aspect polarity classification. Furthermore, these pipeline approaches ignored the correlations between the two separate tasks, which was found to be important for the tasks (Hu et al., 2019). Therefore, many recent researchers handled the ACSA task in a joint way.

2.2. Joint Approach

Many recent researchers handled the ACSA task in a joint way by the classification method. Firstly, Schmitt et al. (2018) jointly modeled the aspect detection and polarity classification in an end-to-end trainable neural network. They added a label (N/A) to the sentiment label space for predicting non-existing aspect categories. Similarly, Wang et al. (2019) used a capsule network structure to predict multiple aspect categories and their polarities.

However, the study (Hu et al., 2019) found that there were only a few words related to the opinion in each aspect, and they proposed a constrained attention network for multi-aspect sentiment analysis. Due to the importance of sentiment-related information associated with the mentioned aspect, the studies (Gu and Zhang, 2022; Fu et al., 2021)

utilized various attention mechanisms to identify such information and obtained good results. The previous methods need to train multiple classifiers separately, and the information among classifiers was not well shared. Therefore, Li et al. (2020a) proposed a novel joint model which contains a shared sentiment prediction layer for ACSA. Similarly, AC-MIMLLN-BERT (Li et al., 2020c) predicted the sentiment of an aspect category by aggregating all sentiment words. Recently, considering the good performance of graph neural networks (GNN) in NLP, Li et al. (2020b); Yang et al. (2020); Cai et al. (2020) introduced GNN to model the correlations between aspect categories or between sentiment words, and achieved good results. Furthermore, Liu et al. (2021) adopted pertained generative model BART (Lewis et al., 2020) for ACSA and outperformed previous models. However, the model requires the pre-trained generative model and some prompt templates.

3. Model

3.1. Problem Formalization

We define some notations and describe the ACSA task. Given a predefined aspect category set $\mathcal{A} = \{a_1, a_2, \dots a_M\}$, sentiment polarity set $\mathcal{P} = \{positive, negative, neutral\}$, and a sentence $\mathbf{x}$ containing N words. Our task is to detect all the mentioned category-sentiment pairs $\mathbf{y}$ from $\mathbf{x}$, formulated as:

$$\mathbf{y} = [y_1, y_2, ..., y_T], \tag{1}$$

where $y_k = (y_k^a, y_k^s)$ is the k^{th} predicted aspect category and aspect sentiment polarity (category-sentiment pair). Consequently, the ACSA can be conceptualized as the search for an optimal sequence $\mathbf{y}$, which maximizes the conditional probability $p(\mathbf{y}|\mathbf{x})$. This probability is computed as:

$$p(\mathbf{y}|\mathbf{x};\theta) = \prod_{t=1}^{T} p(y_t^a|\mathbf{y}_{1:t-1}^a, \mathbf{x};\theta) p(y_t^s|y_t^a, \mathbf{x};\theta), \tag{2}$$

where $\mathbf{y}_{1:t-1}^a$ denotes a sequence $[y_1^a, y_2^a, ..., y_{t-1}^a]$. This indicates that y_t^a is associated with not only the given sentence $\mathbf{x}$ but also the preceding aspect categories. And θ denotes all model parameters.

3.2. Model Architecture

An overview of our proposed model is shown in Figure 2. It consists of two parts: **Sentence Encoder** and **Decoder**. We first use BERT as an encoder. In the decoder, a coverage mechanism and a hierarchical generative mechanism are introduced to generate category-sentiment pair sequences.

3.2.1. Sentence Encoder

A sentence $\mathbf{x}$ in a review is composed of N words, which is formulated as:

$$\mathbf{x} = [w_1, w_2, ..., w_N], \tag{3}$$

where w_i denotes i^{th} word in the sentence. The efficacy of the pre-trained BERT model (Devlin et al., 2018) has been extensively demonstrated in numerous natural language processing (NLP) tasks. Therefore we employ BERT to encode $\mathbf{x}$ and output the context-aware representations $\mathbf{H} = [\mathbf{h}_{CLS}, \mathbf{h}_1, \mathbf{h}_2, ..., \mathbf{h}_N, \mathbf{h}_{SEP}]$. It is important to note that two special tokens, namely [CLS] and [SEP], are inserted at the start and end of each input sentence, respectively. Then we adopt hidden state $\mathbf{h}_{CLS} \in \mathbb{R}^d$ to obtain the initial hidden state of the decoder, which is computed by:

$$\mathbf{s}_0 = \mathbf{W}_0 \mathbf{h}_{CLS} + \mathbf{b}_0, \tag{4}$$

where $\mathbf{W}_0$ and $\mathbf{b}_0$ are the linear transformation weight and bias.

3.2.2. Decoder

The probability of generating the t^{th} aspect category y_t^a is defined as:

$$p(y_t^a|\mathbf{y}_{1:t-1}^a, \mathbf{x}) = softmax(\mathbf{W}_1 \mathbf{s}_t + b_1). \tag{5}$$

where $\mathbf{y}_{1:t-1}^a$ are previous generated aspect categories. The hidden state $\mathbf{s}_t$ of the decoder is computed by:

$$\mathbf{s}_t = f(\mathbf{s}_{t-1}, g(y_{t-1}^a), \mathbf{c}_t), \tag{6}$$

where y_{t-1}^a is the predicted aspect category at time step $t-1$, and $g(y_{t-1}^a)$ is the embedding of y_{t-1}^a,the activation function $f(.)$ is a Gated Recurrent Unit (GRU) (Cho et al., 2014). Usually, in the seq2seq architecture, the context vector $\mathbf{c}_t$ is the weighted sum of the encoder outputs $\mathbf{H}$ (Bahdanau et al., 2014), which is formulated by:

$$\mathbf{c_t} = \sum_{j=1}^{N+2} a_{t,j} \mathbf{h}_j, \tag{7}$$

where $\mathbf{h}_j \in \mathbf{H}$ is the encoder's outputs for word w_j, and $a_{t,j}$ indicates the attention weight for the j^{th} word of the source text at the t time step of the decoder, which is computed by the attention mechanism (Bahdanau et al., 2014):

$$a_{t,j} = \frac{\exp(e_{t,j})}{\sum_{k=1}^{N+2} \exp(e_{t,k})}, \tag{8}$$

where $e_{t,j}$ is computed by:

$$e_{t,j} = \mathbf{v}_a^T tanh(\mathbf{W}_a \mathbf{s}_{t-1} + \mathbf{U}_a \mathbf{h}_j + \mathbf{b}_a), \tag{9}$$

where $\mathbf{W}_a$, $\mathbf{U}_a$, $\mathbf{v}_a$ and $\mathbf{b}_a$ are learnable parameters and $\mathbf{s}_{t-1}$ is the hidden state of the decoder at time step $t-1$.

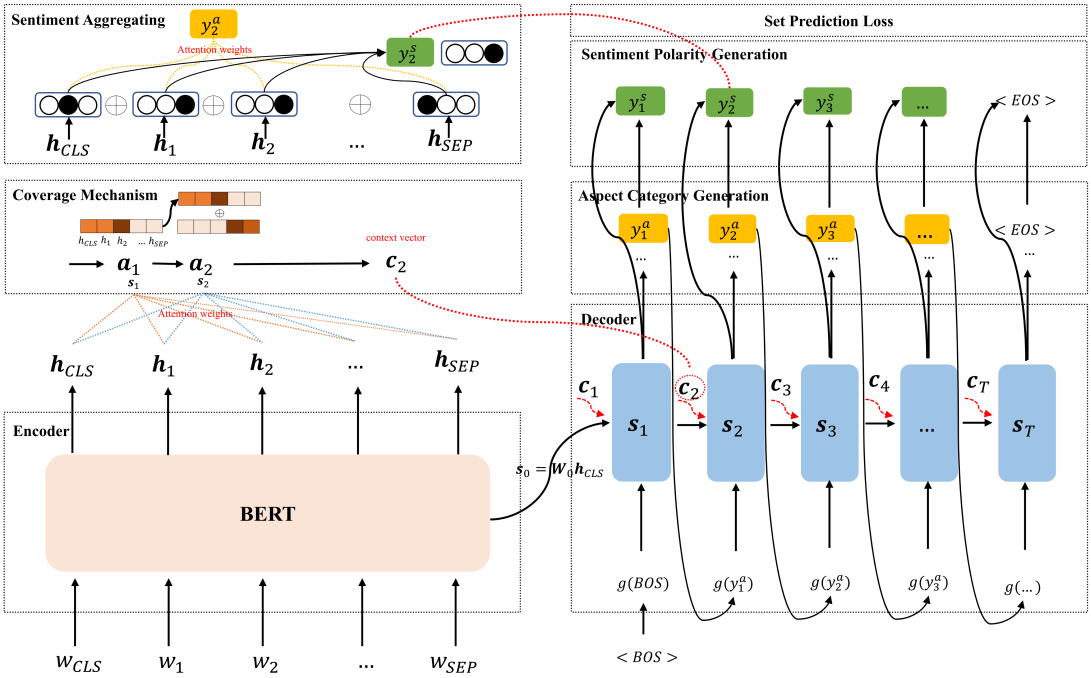

Figure 2: Model Architecture.

Coverage Mechanism As mentioned above, a sentence usually contains one or more category-sentiment pairs. In order to alleviate the missing predictions, we introduce a coverage value (Tu et al., 2016), which can memorize the part covered by previous time steps in the sentence. According to the coverage value, the decoder will increase the attention weight for the words that have previously received less attention and decrease the attention weight for the words that have previously received more attention. Specifically, we rewrite Equation (9) as:

$$e_{t,j} = \mathbf{v}_a^T tanh(\mathbf{W}_a \mathbf{s}_{t-1} + \mathbf{U}_a \mathbf{h}_j + \mathbf{m}_a \tilde{c}_{t-1,j} + \mathbf{b}_a), \tag{10}$$

where $\mathbf{m}_a$ is the weight vector, and $\tilde{c}_{t-1,j}$ is the coverage value of word w_j at time step $t-1$ of the decoder, which is defined as:

$$\tilde{c}_{t-1,j} = \sum_{k=1}^{t-1} a_{k,j}. \tag{11}$$

Intuitively, $\tilde{c}_{t-1,j}$ denotes the degree of coverage derived by word w_j that has received the sum of attention weights at decoder time step $t-1$. A larger value means attention has been paid to w_j by the decoder in the previous time steps. This mechanism simply lowers the attention weights for all previously attended words and increases attention to the remaining words. It is widely utilized in machine translation. The decoder thoroughly analyzes the complete source text to generate the target text, ensuring no information is omitted from the source text. Lastly, the coverage mechanism ensures that the decoder's attention is distributed across various words at different time steps, preventing it from repeatedly focusing on the same words. As a result, the model can accurately generate a greater number of aspect categories from the source text.

Hierarchical Generation Mechanism Considering that different aspect categories should focus on different sentiment words, the polarity of the aspect category should be the aggregation of the polarities of these sentiment words. Inspired by Li et al. (2020c), we design a hierarchical generative mechanism. Specifically, for time step t, the decoder firstly generates aspect category y_t^a by Equation (5). Then we compute the aspect-aware attention weight between the predicted aspect category y_t^a and source word w_j by:

$$a'_{t,j} = \frac{\exp(e'_{t,j})}{\sum_{k=1}^{N+2} \exp(e'_{t,k})}, \tag{12}$$

where $e'_{t,j}$ is computed by:

$$e'_{t,j} = \mathbf{v}_s^T tanh(\mathbf{W}_s g(y_t^a) + \mathbf{U}_s \mathbf{h}_j + \mathbf{b}_s). \tag{13}$$

As mentioned above, the polarity of an aspect should be the aggregation of the polarities of the sentiment words it emphasizes (Li et al., 2020c). Specifically, for word w_j, we predict its polarity by encoder's output $\mathbf{h}_j$:

$$\mathbf{p}_j = \mathbf{W}_{p2}(ReLU(\mathbf{W}_{p1}\mathbf{h}_j + \mathbf{b}_{p1}) + \mathbf{b}_{p2}), \tag{14}$$

where $\mathbf{p}_j \in \mathbb{R}^3$ represents the sentiment predictions of w_j belongs to $\{positive, negative, neutral\}$, respectively. Then we obtain the aspect category polarity by aggregating the word sentiment predictions based on the aspect-aware attention weight. For aspect category y_t^a, its probability of sentiment polarity is computed by:

$$
\begin{aligned}
p(y_t^s|y_t^a, \mathbf{x}) &= softmax(\theta \mathbf{p}_t^1 + (1-\theta)\mathbf{p}_t^2), \\
\mathbf{p}_t^1 &= \sum_{j=1}^{N+2} \mathbf{p}_j a_{t,j}^{'}, \\
\mathbf{p}_t^2 &= tanh(\mathbf{W}_{p3}\mathbf{s}_t + \mathbf{b}_{p3}),
\end{aligned}
\tag{15}
$$

where $a_{t,j}^{'}$ is computed by Equation (12), and $\mathbf{p}_t^2 \in \mathbb{R}^3$ represents the sentiment predictions by $\mathbf{s}_t$. And $\theta \in (0, 1)$ is a learnable parameter.

3.3. Model Optimization

The main difficulty of training is to score the predicted pairs with respect to the ground truths. In this scenario, it is not proper to apply the cross-entropy loss function to measure the difference between two sets, since cross-entropy loss is sensitive to the permutation of the predictions. Inspired by Sui et al. (2023), we propose a set prediction loss that can produce an optimal bipartite matching between predicted and ground truth pairs. Generally, A typical bipartite matching loss computing mainly includes 2 steps (Sui et al., 2023): finding an optimal matching and calculating the loss. After generating N predictions, to find the optimal matching, we first search for a permutation π^* with the lowest cost:

$$
\pi^* = \underset{\pi \in \mathcal{O}_N}{\operatorname{argmin}} \sum_{i=1}^{N} C_{match}(y_i, p_{\pi(i)}), \tag{16}
$$

where $\mathcal{O}_N$ is the space of all N-length permutations, and $C_{match}(.)$ is the matching cost function between ground truths and predicted pairs, which is computed by:

$$
C_{match}(y_i, p_{\pi(i)}) = -\mathbb{I}_{y_i^a \neq \phi}[p_{\pi(i)}^a(y_i^a) + p_{\pi(i)}^s(y_i^s)], \tag{17}
$$

where $p_{\pi(i)}^a, p_{\pi(i)}^s$ are aspect and sentiment probability distribution and computed by Equation(5,15), y_i^a, y_i^s are target aspect and sentiment, respectively. This optimal assignment π^* is computed efficiently by the Hungarian algorithm (Kuhn, 1955). The second step involves computing the loss function for all pairs identified in the preceding step. We define the loss function as follows:

$$
\mathcal{L} = -\sum_{i=1}^{N} [\log p_{\pi^*(i)}^a(y_i^a) + \log p_{\pi^*(i)}^s(y_i^s)]. \tag{18}
$$

Dataset		#Pos	#Neg	#Neu	#Sen
MAMS	Train	2170	2343	3465	3549
	Test	245	263	393	400
Rest	Train	2177	839	500	2891
	Test	657	222	94	767
SRest	Test	379	136	80	595
MRest	Test	278	86	14	172

Table 1: Statistics of the experimental datasets. #Pos, #Neg, and #Neu mean the number of positive, negative, and neutral aspect categories on datasets, respectively. #Sen denotes the number of sentences on datasets.

4. Experiments and Analysis

In this section, we will evaluate our proposed model on four real-world datasets. We first introduce the datasets, evaluation metrics, baseline methods, and experimental settings and then compare our method with the baseline methods. Finally, we provide an elaborate analysis and discussion of experimental results.

4.1. Datasets

We evaluate our model on four datasets, and the statistics of the datasets are shown in Table 1.

- **MAMS** was proposed by Jiang et al. (2019), which is a large-scale Multi-Aspect Multi-Sentiment (MAMS) dataset, in which each sentence contains at least two different aspect categories with different sentiment polarities.

- **Rest** was constructed by SemEval-2014 Task4 (Pontiki et al., 2014), which has been widely used in previous studies (Fu et al., 2021; Wang et al., 2019; Li et al., 2020c). And we removed the data that contains some information of conflicting polarity.

- **SRest** and **MRest** have the same training and validation sets as Rest dataset. However, the data only containing **more than one category-sentiment pairs** can be selected for the test set on MRest dataset, and the data only containing **one category-sentiment pair** can be selected for the test set on SRest dataset.

4.2. Compared Methods

We compare our proposed model with classification and generative model baselines. To ensure the fairness of the comparison, we implement some baselines without source code and repeated the experiment three times on the dataset.

4.2.1. Classification Baselines

We have selected several classic classification baselines, they adopted BERT, CNN, and LSTM as encoders, respectively.

- **AddOneDim-LSTM** (Schmitt et al., 2018) jointly detected aspect categories and classifies their polarity in an end-to-end trainable LSTM. This model added a label (N/A) to the sentiment label space for predicting non-existing aspect categories. **AddOneDim-CNN** and **AddOneDim-BERT** are similar to AddOneDim-LSTM and replaces LSTM with textCNN (Kim, 2014) and BERT (Devlin et al., 2018) as the encoder, respectively.
- **AS-Capsules** (Wang et al., 2019) is an aspect-level sentiment capsules model, which is capable of performing aspect detection and sentiment classification simultaneously[1].
- **AC-MIMLLN-BERT** (Li et al., 2020c) is a multi-instance and multi-label learning network for aspect-category sentiment analysis, which first predicted the sentiments of each word from the source text, then found the key words for the aspect categories, finally obtained the sentiments of the aspect categories by aggregating the sentiments of the key words[2].
- **MSS** (Shi et al., 2023), based on the graph convolutional network (GCN), is a novel unified framework to handle all defined sub-tasks for aspect-based sentiment analysis.

4.2.2. Generation Baselines

We also implement two classic generative models based on Seq2Seq architecture.

- **Seq2Seq-att** is a classic generative model proposed by Sutskever et al. (2014) and introduces bahdanau attention mechanism (Bahdanau et al., 2014). We implement Seq2Seq-att model based on open-sourced code[3]. In our implementation, the output of the decoder is used separately to predict the aspect category and its sentiment polarity.
- **BART-generation** (Liu et al., 2021) is a pre-trained BART model for ACSA.

We further develop two variants of the proposed model. **SGM** is a basic generative model with a BERT encoder and GRU decoder. In the decoder, we feed separately concatenation of decoder output $\mathbf{s}_t$ into two full-connection layers to predict the aspect category and its sentiment polarity simultaneously. **CSGM** is similar to SGM on architecture but integrates the coverage mechanism. **HCGSM+SL** is our complete model and uses the set prediction loss to train the model. In addition, other variants and baseline models use cross-entropy loss to train the models.

4.3. Implementation Details

We implement the baseline models (Seq2Seq-att, BART-generation, AddOneDim-LSTM, AddOneDim-CNN and AddOneDim-BERT). And all hyper-parameters of the models are tuned on the validation dataset by using grid search and early stopping. Our models are developed by Pytorch (Paszke et al., 2019). We set the initial learning rate to $5e-5$ for the decoder and $3e-5$ for the encoder, adopt the dropout strategy to avoid overfitting, and the dropout rate is 0.5. To make a fair comparison, our models and the BERT-based baselines adopt BERT-base-uncased[4] as an encoder, the detailed BERT-base-uncased model settings refer to Devlin et al. (2018). We apply gradient clipping to prevent exploding gradient and set it to 1. In the training phase, AdamW (Loshchilov and Hutter, 2017) is used to optimize the model with a batch size of {16,24,32,48}, and our models use the same hyper-parameter settings. During testing, we use greedy search as the decoding algorithm[5].

We adopt Precision (P), Recall (R), and F1-score (F1) as evaluation metrics that are calculated by comparing the gold category-sentiment pair, and the F1-score is micro-F1. Note that only if the predictions of aspect category and sentiment polarity are identical to the ground truth, the results are treated as correct. Finally, to reduce the randomness of results, we run all models three times and report the average results on the test datasets.

4.4. Performance Comparison

Table 2 shows the results of different models on four datasets. In all classification baselines, the models using BERT as the encoder obtain better results than other classification methods. In particular, the F1 of AC-MIMLLN-BERT, AddOneDim-BERT, and MMS on all datasets far exceed other classification models. This indicates that incorporating BERT can effectively improve the performance of classification models.

In all generative baselines, it can be seen from Table 2 that the pre-trained generation model BART-

[1] https://github.com/thuwyq/WWW19-AS-Capsules
[2] https://github.com/l294265421/AC-MIMLLN
[3] https://github.com/bentrevett/pytorch-seq2seq
[4] https://huggingface.co/bert-base-uncased
[5] Our source code is available at https://github.com/sygogo/HCSGM-ACSA

Category	Models	MAMS			Rest		
		P	R	F1	P	R	F1
Classsification	AddOneDim-LSTM †	58.742	59.563	59.150	72.349	71.326	71.834
	AddOneDim-TextCNN †	56.469	56.937	56.702	64.902	67.523	66.187
	AddOneDim-BERT †	73.246	71.809	72.520	79.943	80.301	80.122
	AS-Capsules	69.250	68.479	68.862	75.799	73.826	74.799
	MSS ⋆	-	-	-	82.520	77.040	79.680
	AC-MIMLLN-BERT	73.228	73.770	73.498	79.829	77.287	78.537
Generation	Seq2Seq-att †	58.239	54.384	56.245	69.696	62.076	65.666
	BART-ACSA †	72.888	71.624	72.250	81.979	80.884	81.428
Our	SGM	68.926	67.037	67.968	81.489	78.040	79.727
	CSGM	68.680	67.703	68.188	81.508	79.274	80.375
	HCSGM	73.882	73.437	73.659	81.438	81.295	81.366
	HCSGM+SL	74.922	73.363	**74.134**	81.578	82.049	**81.813**
Category	Models	SRest			MRest		
		P	R	F1	P	R	F1
Classsification	AddOneDim-LSTM †	66.284	72.325	69.173	85.057	69.753	76.649
	AddOneDim-TextCNN †	58.301	66.499	62.131	78.334	69.136	73.448
	AddOneDim-BERT †	75.488	82.073	78.643	88.652	77.513	82.709
	AS-Capsules	70.549	73.949	72.209	85.904	73.633	79.296
	AC-MIMLLN-BERT	73.166	76.303	74.701	90.735	76.896	83.244
Generation	Seq2Seq-att †	65.322	66.387	65.850	79.773	55.291	65.313
	BART-ACSA †	77.760	81.681	79.672	89.851	79.630	84.432
Our	SGM	78.162	80.392	79.261	87.897	74.339	80.551
	CSGM	78.103	81.120	79.583	87.924	76.367	81.739
	HCSGM	77.545	82.185	79.798	88.638	79.894	84.039
	HCSGM+SL	77.734	82.689	**80.135**	88.610	81.041	**84.657**

Table 2: Comparisons of baselines performances in ACSA. The evaluation results in terms of micro-Precision (P,%) micro-Recall (R,%) and micro-F1 (F1,%), and the baselines marked † are our implementations, ⋆ refers to citing from Shi et al. (2023). We run all models three times and report the average results on the test sets. The best F1 results are bold.

generation has achieved better results in performance, and its performance far exceeds that of Seq2Seq-att and classification models. Furthermore, our complete model achieves an obvious improvement in F1 over the second-best Model.

4.5. Ablation Study

In this section, we will further analyze the influence of the coverage mechanism, hierarchical generative mechanism, and set prediction loss, respectively.

4.5.1. Impact of Coverage Mechanism

To further analyze the effectiveness of the coverage mechanism, we conducted an ablation study. First of all, it can be seen from Table 2 that compared with the standard attention decoder (SGM), after the introduction of the coverage mechanism, the model performance has been greatly improved in the ACSA task. As mentioned above, the coverage mechanism can help the model recall more aspect categories, so we conducted experiments on the ACD task, and the results are shown in Table 3. It can be seen that after the introduction of the coverage mechanism, F1 and accuracy have been greatly improved, especially on the MRest dataset, the improvement is more obvious.

4.5.2. Impact of Hierarchical Generative Mechanism

In this section, we study the ability of the models to predict sentiment polarity. We first define a new metric: **P**olarity **P**rediction **E**rror **R**ate (PPER), which is computed by:

$$PPER = \frac{\#FP_p}{\#GP}, \tag{19}$$

where $\#GP$ denotes the number of ground truth pairs, and $\#FP_p$ denotes the number of false predicted pairs due to polarity prediction error. For example, the ground truth is

$$\{(food, positive), (service, positive)\}, \tag{20}$$

and predicted pairs are

$$\{(food, positive), \underbrace{(service, negative)}_{polarity\ error}, \underbrace{(price, positive)}_{category\ error}\}. \tag{21}$$

Model	MAMS		Rest		SRest		MRest	
	F1	Acc	F1	Acc	F1	Acc	F1	Acc
SGM	**92.355**	**93.979**	95.230	95.976	95.998	96.919	93.367	92.713
CSGM	92.231	93.865	**95.716**	**96.367**	**96.181**	**97.053**	**94.601**	**93.992**

Table 3: Performance of ACD on four datasets.The evaluation results in terms micro-F1 (F1,%) and accuracy (Acc,%).

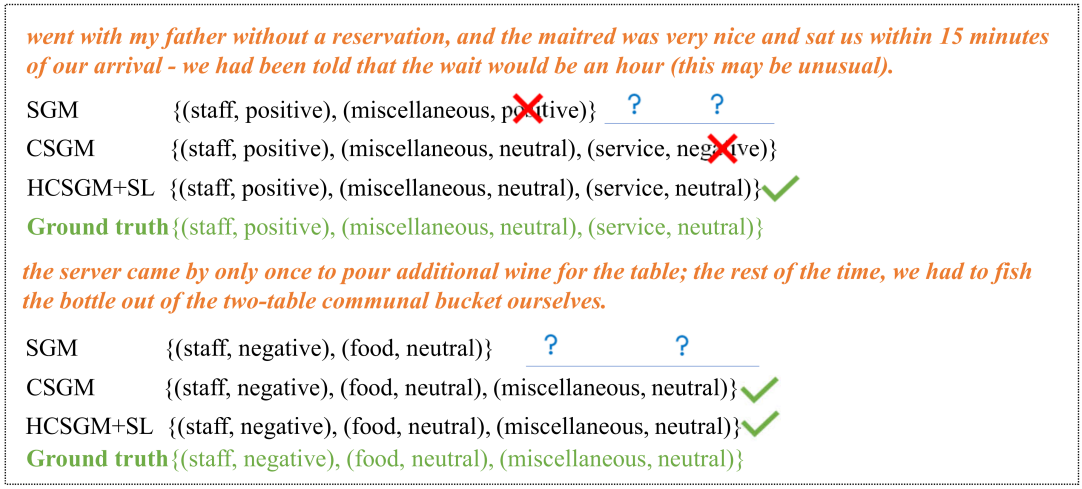

Figure 3: Case study on MAMS dataset. False prediction pairs are marked with "×" and missing pairs are marked with "?".

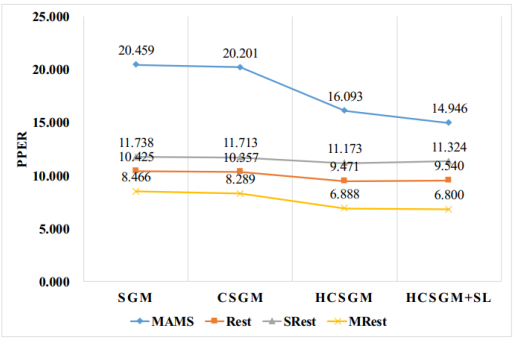

Figure 4: Comparsion of PPER (%) on four datasets.

In this case, $\#GP = 2$, the predicted pair $(service, negative)$ is a false predicted pair due to sentiment polarity prediction error, and $(price, positive)$ is also a false predicted pair due to aspect category prediction error. Therefore, $\#FP_p = 1$ and $PPER = 0.5$. It can be seen that PPER represents the prediction ability of the model for sentiment polarity. When PPER is lower, the prediction ability of the model is better.

Then we compute $PPER$ of four models on four datasets, and the experimental result is shown in Figure 4. It can be seen that PPER of introducing the hierarchical generative mechanism decreases significantly on every dataset due to the introduction of the hierarchical generative mechanism, especially on the MAMS dataset. On the contrary, the introduction of the coverage mechanism and set loss has basically no obvious impact on PPER.

4.5.3. Impact of Set Loss

Table 2 demonstrates that adding set prediction loss significantly improves F1, especially on datasets with more than one aspect category, such as MRest and MAMS. However, the increase in F1 is smaller on datasets with only one aspect category, such as SRest.

4.6. Case Study

Finally, we present a case study on MAMS dataset by different models. Figure 3 presents the cases of SGM, CSGM, and HCSGM, respectively. For case one, we can see that SGM generates two correct category-sentiment pairs and misses one pair `(service, neutral)`. After introducing the coverage mechanism, the model CSGM accurately generates three aspect categories, but the predicted polarity of the last aspect category is still wrong. Finally, the model HCSGM using the hierarchical generative and coverage mechanism completely correctly generates all category-sentiment pairs. The second case also shows the same result.

It can be seen from the case study that the coverage mechanism alleviates the missing predic-

tion of aspect categories, and the hierarchical generative mechanism correctly predicts the polarity of these aspect categories and finally makes the model achieve better performance.

5. Conclusions

In this paper, we propose a hierarchical sequence-to-set model with a coverage mechanism for ACSA. It includes a BERT-based encoder and a GRU decoder, meanwhile, a coverage mechanism is introduced to avoid the missing predictions of category-sentiment pairs. Furthermore, in the decoder, we design a hierarchical generative mechanism to ensure that different aspect categories can focus on different sentiment words. The extended experiments show that our model achieves better performance on four datasets. In addition, we also perform an ablation study, which proves the effectiveness of the coverage and hierarchical generative mechanism.

6. Bibliographical References

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*.

Caroline Brun, Julien Perez, and Claude Roux. 2016. Xrce at semeval-2016 task 5: Feedbacked ensemble modeling on syntactico-semantic knowledge for aspect based sentiment analysis. In *Proceedings of the 10th international workshop on semantic evaluation (SemEval-2016)*, pages 277–281.

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.

Kyunghyun Cho, Bart Van Merriënboer, Dzmitry Bahdanau, and Yoshua Bengio. 2014. On the properties of neural machine translation: Encoder-decoder approaches. *arXiv preprint arXiv:1409.1259*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Yujie Fu, Jian Liao, Yang Li, Suge Wang, Deyu Li, and Xiaoli Li. 2021. Multiple perspective attention based on double bilstm for aspect and sentiment pair extract. *Neurocomputing*, 438:302–311.

Ping Gu and Zhipeng Zhang. 2022. Dual-attention based joint aspect sentiment classification model. In *Web Engineering: 22nd International Conference, ICWE 2022, Bari, Italy, July 5–8, 2022, Proceedings*, pages 252–267. Springer.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9(8):1735–1780.

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.

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.

Yoon Kim. 2014. Convolutional neural networks for sentence classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1746–1751.

Harold W Kuhn. 1955. The hungarian method for the assignment problem. *Naval research logistics quarterly*, 2(1-2):83–97.

Ayush Kumar, Sarah Kohail, Amit Kumar, Asif Ekbal, and Chris Biemann. 2016. Iit-tuda at semeval-2016 task 5: Beyond sentiment lexicon: Combining domain dependency and distributional semantics features for aspect based sentiment analysis. In *Proceedings of the 10th international workshop on semantic evaluation (SemEval-2016)*, pages 1129–1135.

Ji-Ung Lee, Steffen Eger, Johannes Daxenberger, and Iryna Gurevych. 2017. Ukp tu-da at germeval 2017: Deep learning for aspect based sentiment detection. *Proceedings of the GSCL GermEval Shared Task on Aspect-based Sentiment in Social Media Customer Feedback*, pages 22–29.

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.

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 *Chinese Computational Linguistics: 19th China National Conference, CCL 2020, Hainan, China, October 30–November 1, 2020, Proceedings 19*, pages 388–400. Springer.

Yuncong Li, Cunxiang Yin, and Sheng-hua Zhong. 2020b. Sentence constituent-aware aspect-category sentiment analysis with graph attention networks. In *Natural Language Processing and Chinese Computing: 9th CCF International Conference*, pages 815–827.

Yuncong Li, Cunxiang Yin, Sheng-hua Zhong, and Xu Pan. 2020c. 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.

Bing Liu. 2012. Sentiment analysis and opinion mining. *Synthesis lectures on human language technologies*, 5(1):1–167.

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.

Ilya Loshchilov and Frank Hutter. 2017. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. 2019. Pytorch: An imperative style, high-performance deep learning library. In *Advances in neural information processing systems*, volume 32.

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. Association for Computational Linguistics.

Sebastian Ruder. 2017. An overview of multi-task learning in deep neural networks. *arXiv preprint arXiv:1706.05098*.

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.

Jingli Shi, Weihua Li, Quan Bai, Yi Yang, and Jianhua Jiang. 2023. Syntax-enhanced aspect-based sentiment analysis with multi-layer attention. *Neurocomputing*, 557:126730.

Dianbo Sui, Xiangrong Zeng, Yubo Chen, Kang Liu, and Jun Zhao. 2023. Joint entity and relation extraction with set prediction networks. *IEEE Transactions on Neural Networks and Learning Systems*.

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. In *Advances in Neural Information Processing Systems*, volume 27. Curran Associates, Inc.

Zhaopeng Tu, Zhengdong Lu, Yang Liu, Xiaohua Liu, and Hang Li. 2016. Modeling coverage for neural machine translation. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 76–85.

Yequan Wang, Aixin Sun, Minlie Huang, and Xiaoyan Zhu. 2019. Aspect-level sentiment analysis using as-capsules. In *The world wide web conference*, pages 2033–2044.

Ying Yang, Bin Wu, Lianwei Li, and Shuyang Wang. 2020. A joint model for aspect-category sentiment analysis with textgcn and bi-gru. In *2020 IEEE Fifth International Conference on Data Science in Cyberspace (DSC)*, pages 156–163. IEEE.

Wenxuan Zhang, Xin Li, Yang Deng, Lidong Bing, and Wai Lam. 2022. A survey on aspect-based sentiment analysis: tasks, methods, and challenges. *IEEE Transactions on Knowledge and Data Engineering*.

Tao Zhou and Kris MY Law. 2022. Semantic relatedness enhanced graph network for aspect category sentiment analysis. *Expert Systems with Applications*, 195:116560.