A Unified Generative Framework for Aspect-Based Sentiment Analysis

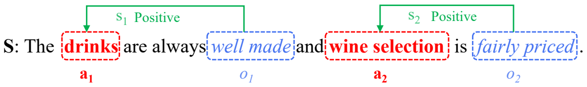
Hang Yan[1,*], Junqi Dai[1,*], Tuo Ji[1], Xipeng Qiu[1,2†], Zheng Zhang[3]
[1]Shanghai Key Laboratory of Intelligent Information Processing, Fudan University
[1]School of Computer Science, Fudan University
[2]Pazhou Lab, Guangzhou, China
[3]New York University
{hyan19,jqdai19,tji19,xpqiu}@fudan.edu.cn
zz@nyu.edu

Abstract

Aspect-based Sentiment Analysis (ABSA) aims to identify the aspect terms, their corresponding sentiment polarities, and the opinion terms. There exist seven subtasks in ABSA. Most studies only focus on the subsets of these subtasks, which leads to various complicated ABSA models while hard to solve these subtasks in a unified framework. In this paper, we redefine every subtask target as a sequence mixed by pointer indexes and sentiment class indexes, which converts all ABSA subtasks into a unified generative formulation. Based on the unified formulation, we exploit the pre-training sequence-to-sequence model BART to solve all ABSA subtasks in an end-to-end framework. Extensive experiments on four ABSA datasets for seven subtasks demonstrate that our framework achieves substantial performance gain and provides a real unified end-to-end solution for the whole ABSA subtasks, which could benefit multiple tasks[1].

1 Introduction

Aspect-based Sentiment Analysis (ABSA) is the fine-grained Sentiment Analysis (SA) task, which aims to identify the aspect term (a), its corresponding sentiment polarity (s), and the opinion term (o). For example, in the sentence "*The drinks are always well made and wine selection is fairly priced*", the aspect terms are "*drinks*" and "*wine selection*", and their sentiment polarities are both "positive", and the opinion terms are "*well made*" and "*fairly priced*". Based on the combination of the a, s, o, there exist seven subtasks in ABSA. We summarize these subtasks in Figure 1. Specifically, their definitions are as follows:


s_1 Positive
s_2 Positive
S: The drinks are always well made and wine selection is fairly priced.
a_1
o_1
a_2
o_2

Subtask	Input	Output	Task Type
Aspect Term Extraction(*AE*)	S	a_1, a_2	Extraction
Opinion Term Extraction(*OE*)	S	o_1, o_2	Extraction
Aspect-level Sentiment Classification(*ALSC*)	S + a_1	s_1	Classification
	S + a_2	s_2	
Aspect-oriented Opinion Extraction(*AOE*)	S + a_1	o_1	Extraction
	S + a_2	o_2	
Aspect Term Extraction and Sentiment Classification(*AESC*)	S	(a_1, s_1), (a_2, s_2)	Extraction & Classification
Pair Extraction(*Pair*)	S	(a_1, o_1), (a_2, o_2)	Extraction
Triplet Extraction(*Triplet*)	S	(a_1, o_1, s_1), (a_2, o_2, s_2)	Extraction & Classification

Figure 1: Illustration of seven ABSA subtasks.

- Aspect Term Extraction(*AE*): Extracting all the aspect terms from a sentence.
- Opinion Term Extraction (*OE*): Extracting all the opinion terms from a sentence.
- Aspect-level Sentiment Classification (*ALSC*): Predicting the sentiment polarities for every given aspect terms in a sentence.
- Aspect-oriented Opinion Extraction (*AOE*): Extracting the paired opinion terms for every given aspect terms in a sentence.
- Aspect Term Extraction and Sentiment Classification (*AESC*): Extracting the aspect terms as well as the corresponding sentiment polarities simultaneously.
- Pair Extraction (*Pair*): Extracting the aspect terms as well as the corresponding opinion terms simultaneously.
- Triplet Extraction (*Triplet*): Extracting all aspects terms with their corresponding opinion terms and sentiment polarity simultaneously.

Although these ABSA subtasks are strongly related, most of the existing work only focus 1∼3 subtasks individually. The following divergences make it difficult to solve all subtasks in a unified framework.

1. *Input:* Some subtasks (*AE*, *OE*, *AESC*, *Pair*

[*] Equal contribution.
[†] Corresponding author.
[1] Code is available at https://github.com/yhcc/BARTABSA.

*Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics
and the 11th International Joint Conference on Natural Language Processing*, pages 2416–2429
August 1–6, 2021. ©2021 Association for Computational Linguistics

and *Triplet*) only take the text sentence as input, while the remained subtasks (*ALSC* and *AOE*) take the text and a given aspect term as input.

2. *Output:* Some tasks (*AE*, *OE*, *ALSC*, *AOE*) only output a certain type from a, s or o, while the remained tasks (*AESC*, *Pair* and *Triplet*) return compound output as the combination of a, s and o.
3. *Task Type:* There are two kinds of tasks: extraction task (extracting aspect and opinion) and classification task (predicting sentiment).

Because of the above divergences, a myriad of previous works only focus on the subset of these subtasks. However, the importance of solving the whole ABSA subtasks in a unified framework remains significant. Recently, several works make attempts on this track. Some methods(Peng et al., 2020; Mao et al., 2021) apply the pipeline model to output the a, s, o from the inside sub-models separately. However, the pipeline process is not end-to-end. Another line follows the sequence tagging method by extending the tagging schema (Xu et al., 2020). However, the compositionality of candidate labels hinders the performance. In conclusion, the existing methods can hardly solve all the subtasks by a unified framework without relying on the sub-models or changing the model structure to adapt to all ABSA subtasks.

Motivated by the above observations, we propose a unified generative framework to address all the ABSA subtasks. We first formulate all these subtasks as a generative task, which could handle the obstacles on the input, output, and task type sides and adapt to all the subtasks without any model structure changes. Specifically, we model the extraction and classification tasks as the pointer indexes and class indexes generation, respectively. Based on the unified task formulation, we use the sequence-to-sequence pre-trained model BART (Lewis et al., 2020) as our backbone to generate the target sequence in an end-to-end process. To validate the effectiveness of our method, we conduct extensive experiments on public datasets. The comparison results demonstrate that our proposed framework outperforms most state-of-the-art (SOTA) models in every subtask.

In summary, our main contributions are as follows:

• We formulate both the extraction task and classification task of ABSA into a unified index generation problem. Unlike previous unified models, our method needs not to design specific decoders for different output types.

• With our re-formulation, all ABSA subtasks can be solved in sequence-to-sequence framework, which is easy-to-implement and can be built on the pre-trained models, such as BART.

• We conduct extensive experiments on four public datasets, and each dataset contains a subset of all ABSA subtasks. To the best of our knowledge, it is the first work to evaluate a model on all ABSA tasks.

• The experimental results show that our proposed framework significantly outperforms recent SOTA methods.

2 Background

2.1 ABSA Subtasks

In this section, we first review the existing studies on single output subtasks, and then turn to studies focusing on the compound output subtasks.

2.1.1 Single Output Subtasks

Some researches mainly focus on the single output subtasks. The *AE*, *OE*, *ALSC* and *AOE* subtasks only output one certain type from a, s or o.

AE Most studies treat *AE* subtask as a sequence tagging problem (Li and Lam, 2017; Xu et al., 2018; Li et al., 2018b). Recent works explore sequence-to-sequence learning on *AE* subtask, which obtain promising results especially with the pre-training language models (Ma et al., 2019; Li et al., 2020).

OE Most studies treat *OE* subtask as an auxiliary task (Wang et al., 2016a, 2017; Wang and Pan, 2018; Chen and Qian, 2020; He et al., 2019). Most works can only extract the unpaired aspect and opinion terms[2]. In this case, opinion terms are independent of aspect terms.

ALSC Tang et al. (2016a) use the long short term memory (LSTM) network to enhance the interactions between aspects and context words. Wang et al. (2016b); Liu and Zhang (2017); Ma et al. (2017); Tay et al. (2018) incorporate the attention mechanism into the LSTM-based neural network models to model relations of aspects and their contextual words. Other model structures such as convolutional neural network (CNN) (Li et al., 2018a; Xue and Li, 2018), gated neural network (Zhang et al., 2016; Xue and Li, 2018), memory neural

[2] It is also referred to as the AE-OE co-Extraction.

network (Tang et al., 2016b; Chen et al., 2017) have also been applied.

AOE This subtask is first introduced by Fan et al. (2019) and they propose the datasets for this subtask. Most studies apply sequence tagging method for this subtask (Wu et al., 2020; Pouran Ben Veyseh et al., 2020).

2.1.2 Compound Output Subtasks

Some researchers pay more attention and efforts to the subtasks with compound output. We review them as follows:

AESC. One line follows pipeline method to solve this problem. Other works utilize unified tagging schema (Mitchell et al., 2013; Zhang et al., 2015; Li et al., 2019) or multi-task learning (He et al., 2019; Chen and Qian, 2020) to avoid the error-propagation problem (Ma et al., 2018). Span-based *AESC* works are also proposed recently (Hu et al., 2019), which can tackle the sentiment inconsistency problem in the unified tagging schema.

Pairs Zhao et al. (2020) propose to extract all (a, o) pair-wise relations from scratch. They propose a multi-task learning framework based on the span-based extraction method to handle this subtask.

Triplet This subtask is proposed by Peng et al. (2020) and gains increasing interests recently. Xu et al. (2020) design the position-aware tagging schema and apply model based on CRF (Lafferty et al., 2001) and Semi-Markov CRF (Sarawagi and Cohen, 2004). However, the time complexity limits the model to detect the aspect term with long-distance opinion terms. Mao et al. (2021) formulate *Triplet* as a two-step MRC problem, which applies the pipeline method.

2.2 Sequence-to-Sequence Models

The sequence-to-sequence framework has been long studied in the NLP field to tackle various tasks (Sutskever et al., 2014; Cho et al., 2014; Vinyals et al., 2015; Luong et al., 2015). Inspired by the success of PTMs (pre-trained models) (Qiu et al., 2020; Peters et al., 2018; Devlin et al., 2019; Brown et al., 2020), Song et al. (2019); Raffel et al. (2020); Lewis et al. (2020) try to pre-train sequence-to-sequence models. Among them, we use the BART (Lewis et al., 2020) as our backbone, while the other sequence-to-sequence pre-training models can also be applied in our architecture to use the pointer mechanism (Vinyals et al., 2015), such as MASS (Song et al., 2019).

BART is a strong sequence-to-sequence pre-trained model for Natural Language Generation (NLG). BART is a denoising autoencoder composed of several transformer (Vaswani et al., 2017) encoder and decoder layers. It is worth noting that the BART-Base model contains a 6-layer encoder and 6-layer decoder, which makes it similar number of parameters[3] with the BERT-Base model. BART is pretrained on denoising tasks where the input sentence is noised by some methods, such as masking and permutation. The encoder takes the noised sentence as input, and the decoder will restore the original sentence in an autoregressive manner.

3 Methodology

Although there are two types of tasks among the seven ABSA subtasks, they can be formulated under a generative framework. In this part, we first introduce our sequential representation for each ABSA subtask. Then we detail our method, which utilizes BART to generate these sequential representations.

3.1 Task Formulation

As depicted in Figure 1, there are two types of tasks, namely the extraction and classification, whose target can be represented as a sequence of pointer indexes and class indexes, respectively. Therefore, we can formulate these two types of tasks in a unified generative framework. We use a, s, o, to represent the aspect term, sentiment polarity,and opinion term, respectively. Moreover, we use the superscript ^{s} and ^{e} to denote the start index and end index of a term. For example, o^s, a^e represent the start index of an opinion term o and the end index of an aspect term a. We use the s^p to denote the index of sentiment polarity class. The target sequence for each subtask is as follows:

- *AE* : $Y = [a_1^s, a_1^e, ..., a_i^s, a_i^e, ...]$,
- *OE* : $Y = [o_1^s, o_1^e, ..., o_i^s, o_i^e, ...]$,
- *AESC* : $Y = [a_1^s, a_1^e, s_1^p, ..., a_i^s, a_i^e, s_i^p, ...]$,
- *Pair*: $Y = [a_1^s, a_1^e, o_1^s, o_1^e, ..., a_i^s, a_i^e, o_i^s, o_i^e, ...]$,
- *Triplet* : $Y = [a_1^s, a_1^e, o_1^s, o_1^e, s_1^p, ..., a_i^s, a_i^e, o_i^s, o_i^e, s_i^p, ...]$,

The above subtasks only rely on the input sentence, while for the *ALSC* and *AOE* subtasks, they also depend on a specific aspect term a. Instead of putting the aspect term on the input side, we put

[3] Because of the cross-attention between encoder and decoder, the number of parameters of BART is about 10% larger than its counterpart of BERT (Lewis et al., 2020).

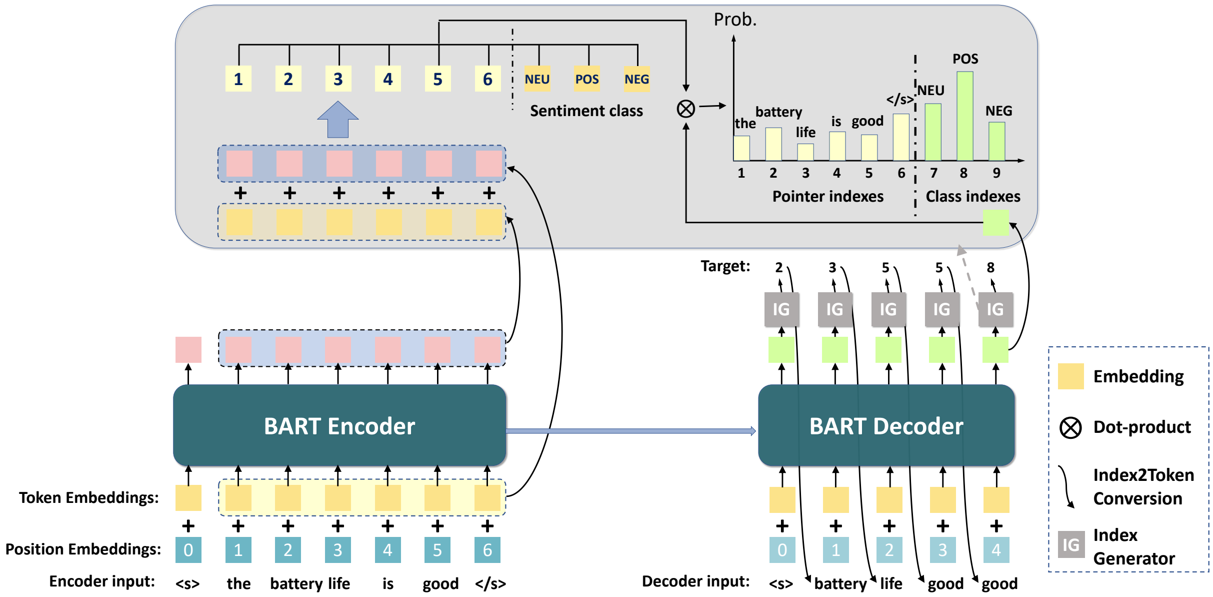
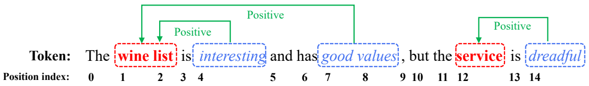


Figure 2: Overall architecture of the framework. This shows an example generation process for the *Triplet* subtask where the source is "*<s>the battery life is good </s>*" and the target is "*2 3 5 5 8 6*"(Only partial decoder sequence is shown where the 6 (</s>) should be the next generation index). The "Index2Token Conversion" converts the index to tokens. Specifically, the pointer index will be converted to its corresponding token in the source text, and the class index will be converted to corresponding class tokens. Embedding vectors in 🟨 boxes are retrieved from same embedding matrix. We use different position embeddings in the source and target for better generation performance.


Positive
Positive
Positive
Token: The wine list is interesting and has good values, but the service is dreadful
Position index: 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14

Subtask	Target Sequence
AE	1, 2, 12, 12, </s>
OE	4, 4, 7, 8, 14, 14, </s>
ALSC	$\underline{1}$, $\underline{2}$, POS, </s>
	$\underline{12}$, $\underline{12}$, POS, </s>
AOE	$\underline{1}$, $\underline{2}$, 4, 4, 7, 8, </s>
	$\underline{12}$, $\underline{12}$, 14, 14, </s>
AESC	1, 2, POS, 12, 12, NEG, </s>
Pair	1, 2, 4, 4, 1, 2, 7, 8, 12, 12, 14, 14, </s>
Triplet	1, 2, 4, 4, POS, 1, 2, 7, 8, POS, 12, 12, 14, 14, POS, </s>

Figure 3: Target sequences for different subtasks. The underlined indexes are given in advance. We convert the sentiment class index to the corresponding class token for better understanding.

them on the target side so that the target sequences are as follows:

- $ALSC : Y = [\underline{a^s}, \underline{a^e}, s^p]$,
- $AOE : Y = [\underline{a^s}, \underline{a^e}, o^s_1, o^e_1, ..., o^s_i, o^e_i, ...]$,

where the underlined tokens are given during inference. Detailed target sequence examples for each subtask are presented in Figure 3.

3.2 Our Model

As our discussion in the last section, all subtasks can be formulated as taking the $X = [x_1, ..., x_n]$ as input and outputting a target sequence $Y = [y_1, ..., y_m]$, where y_0 is the start-of-the-sentence token. Therefore, different ABSA subtasks can be formulated as:

$$P(Y|X) = \prod_{t=1}^{m} P(y_t|X, Y_{<t}). \qquad (1)$$

To get the index probability distribution $P_t = P(y_t|X, Y_{<t})$ for each step, we use a model composed of two components: (1) **Encoder**; (2) **Decoder**.

Encoder The encoder part is to encode X into vectors $\mathbf{H}^e$. We use the BART model, therefore, the start of sentence (<s>) and the end of sentence (</s>) tokens will be added to the start and end of X, respectively. We ignore the <s> token in our equations for simplicity. The encoder part is as follows:

$$\mathbf{H}^e = \mathrm{BARTEncoder}([x_1, ..., x_n]), \qquad (2)$$

where $\mathbf{H}^e \in \mathbb{R}^{n \times d}$, and d is the hidden dimension.

Decoder The decoder part takes the encoder outputs $\mathbf{H}^e$ and previous decoder outputs $Y_{<t}$ as inputs to get P_t. However, the $Y_{<t}$ is an index sequence. Therefore, for each y_t in $Y_{<t}$, we first need to use the following Index2Token module to conduct a

Dataset		14res				14lap				15res				16res				Subtasks
		#s	#a	#o	#p	#s	#a	#o	#p	#s	#a	#o	#p	#s	#a	#o	#p	
$\mathcal{D}_{17}$	train	3044	3699	3484	-	3048	2373	2504	-	1315	1199	1210	-	-	-	-	-	*AE*, *OE*, *ALSC*, *AESC*
	test	800	1134	1008	-	800	654	674	-	685	542	510	-	-	-	-	-	
$\mathcal{D}_{19}$	train	1627	2643	-	-	1158	1634	-	-	754	1076	-	-	1079	1512	-	-	*AOE*
	test	500	865	-	-	343	482	-	-	325	436	-	-	329	457	-	-	
$\mathcal{D}_{20a}$	train	1300	-	-	2145	920	-	-	1265	593	-	-	923	842	-	-	1289	*AE*, *OE*, *ALSC*, *AOE*, *AESC*, *Pair*, *Triplet*
	dev	323	-	-	524	228	-	-	337	148	-	-	238	210	-	-	316	
	test	496	-	-	862	339	-	-	490	318	-	-	455	320	-	-	465	
$\mathcal{D}_{20b}$	train	1266	-	-	2338	906	-	-	1460	605	-	-	1013	857	-	-	1394	*AE*, *OE*, *ALSC*, *AOE*, *AESC*, *Pair*, *Triplet*
	dev	310	-	-	577	219	-	-	346	148	-	-	249	210	-	-	339	
	test	492	-	-	994	328	-	-	543	148	-	-	485	326	-	-	514	

Table 1: The statistics of four datasets, where the #s, #a, #o, #p denote the numbers of sentences, aspect terms, opinion terms, and the $<a$, $o>$ pairs, respectively. We use "-" to denote the missing data statistics of some datasets. The "Subtasks" column refers to the ABSA subtasks that can be applied on the corresponding dataset.

conversion

$$\hat{y}_t = \begin{cases} X_{y_t}, & \text{if } y_t \text{ is a pointer index,} \\ C_{y_t - n}, & \text{if } y_t \text{ is a class index,} \end{cases} \tag{3}$$

where $C = [c_1, ..., c_l]$ is the class token list[4].

After that, we use the BART decoder to get the last hidden state

$$\mathbf{h}_t^d = \mathrm{BARTDecoder}(\mathbf{H}^e; \hat{Y}_{<t}), \tag{4}$$

where $\mathbf{h}_t^d \in \mathbb{R}^d$. With $\mathbf{h}_t^d$, we predict the token probability distribution P_t as follows:

$$\mathbf{E}^e = \mathrm{BARTTokenEmbed}(X), \tag{5}$$

$$\hat{\mathbf{H}}^e = \mathrm{MLP}(\mathbf{H}^e), \tag{6}$$

$$\bar{\mathbf{H}}^e = \alpha \hat{\mathbf{H}}^e + (1 - \alpha)\mathbf{E}^e, \tag{7}$$

$$\mathbf{C}^d = \mathrm{BARTTokenEmbed}(C), \tag{8}$$

$$P_t = \mathrm{Softmax}([\bar{\mathbf{H}}^{\mathbf{e}}; \mathbf{C}^d]\mathbf{h}_t^d), \tag{9}$$

where $\mathbf{E}^e, \mathbf{H}^e, \hat{\mathbf{H}}^e, \bar{\mathbf{H}}^e \in \mathbb{R}^{n \times d}$; $\mathbf{C}^d \in \mathbb{R}^{l \times d}$; and $P_t \in \mathbb{R}^{(n+l)}$ is the final distribution on all indexes.

During the training phase, we use the teacher forcing to train our model and the negative log-likelihood to optimize the model. Moreover, during the inference, we use the beam search to get the target sequence Y in an autoregressive manner. After that, we need to use the decoding algorithm to convert this sequence into the term spans and sentiment polarity. We use the *Triplet* task as an example and present the decoding algorithm in Algorithm 1, the decoding algorithm for other tasks are much depicted in the Supplementary Material.

Algorithm 1 Decoding Algorithm for the *Triplet* Subtask

```
Input: Number of tokens in the input sentence
    $n$, target sequence $Y = [y_1, ..., y_m]$ and $y_i \in [1, n + |C|]$
Output: Target span set $L = \{(a_1^s, a_1^e, o_1^s, o_1^e, s_1), ..., (a_i^s, a_i^e, o_i^s, o_i^e, s_i), ...\}$
$L = \{\}, e = [], i = 1$
while $i <= m$ do
    $y_i = Y[i]$
    if $y_i > n$ then
        $L.add((e, C_{y_i - n}))$
        $e = []$
    else
        $e.append(y_i)$
    end if
    $i += 1$
end while
return $L$
```

4 Experiments

4.1 Datasets

We evaluate our method on four ABSA datasets. All of them are originated from the Semeval Challenges (Pontiki et al., 2014a,b,c), where only the aspect terms and their sentiment polarities are labeled.

The first dataset($\mathcal{D}_{17}$[5]) is annotated by Wang et al. (2017), where the unpaire opinion terms are labeled. The second dataset($\mathcal{D}_{19}$) is annotated by Fan et al. (2019), where they pair opinion terms with

[4] In our implement, $y_t \in [1, n + l]$. The x_1 has the pointer index 1.

[5] Each dataset only contains a subset of all ABSA subtasks. We use the published year of the dataset to distinguish them.

Baselines	E2E	Task Formulation	Backbone	Datasets	*AE*	*OE*	*ALSC*	*AOE*	*AESC*	*Pair*	*Triplet*
SPAN-BERT	-	Span.Extraction	BERT	$\mathcal{D}_{17}$	✓	-	✓	-	✓	-	-
IMN-BERT	✓	Seq.Tagging	BERT	$\mathcal{D}_{17}$	✓	✓	✓	-	✓	-	-
RACL-BERT	-	Seq.Tagging	BERT	$\mathcal{D}_{17}$	✓	✓	✓	-	✓	-	-
IOG	✓	Seq.Tagging	LSTM	$\mathcal{D}_{19}$	-	-	-	✓	-	-	-
LOTN	✓	Seq.Tagging	LSTM	$\mathcal{D}_{19}$	-	-	-	✓	-	-	-
ONG	✓	Seq.Tagging	BERT	$\mathcal{D}_{19}$	-	-	-	✓	-	-	-
RINANTE+	-	Seq.Tagging	LSTM+CRF	$\mathcal{D}_{20a}$,$\mathcal{D}_{20b}$	✓	✓	✓	-	✓	✓	✓
CMLA+	-	Seq.Tagging	Attention	$\mathcal{D}_{20a}$,$\mathcal{D}_{20b}$	✓	✓	✓	-	✓	✓	✓
Li-unified+	-	Seq.Tagging	LSTM	$\mathcal{D}_{20a}$,$\mathcal{D}_{20b}$	✓	✓	✓	-	✓	✓	✓
Peng-two-stage	-	Seq.Tagging	LSTM+GCN	$\mathcal{D}_{20a}$,$\mathcal{D}_{20b}$	✓	✓	✓	-	✓	✓	✓
JET-BERT	✓	Seq.Tagging	BERT	$\mathcal{D}_{20a}$,$\mathcal{D}_{20b}$	✓	✓	✓	-	✓	✓	✓
Dual-MRC	-	Span.MRC	BERT	$\mathcal{D}_{17}$,$\mathcal{D}_{19}$,$\mathcal{D}_{20a}$,$\mathcal{D}_{20b}$	✓	-	✓	✓	✓	✓	✓
Ours	✓	Span.Generation	BART	$\mathcal{D}_{17}$,$\mathcal{D}_{19}$,$\mathcal{D}_{20a}$,$\mathcal{D}_{20b}$	✓	✓	✓	✓	✓	✓	✓

Table 2: The baselines of our experiments. To further demonstrate that our proposed method is a real unified end-to-end ABSA framework, we present our work in the last row. “E2E” is short for End-to-End, which means the model should output all the subtasks’ results synchronously rather than requiring any preconditions, e.g., pipeline methods. The “Datasets” column refers to the datasets that this baseline is conducted.

corresponding aspects. The third dataset($\mathcal{D}_{20a}$) is from Peng et al. (2020). They refine the data in $<a, o, s>$ triplet form. The fourth dataset($\mathcal{D}_{20b}$) from Xu et al. (2020) is the revised variant of Peng et al. (2020), where the missing triplets with overlapping opinions are corrected. We present the statistics for these four datasets in Table 1.

4.2 Baselines

To have a fair comparison, we summarize top-performing baselines of all ABSA subtasks. Given different ABSA subtasks, datasets, and experimental setups, existing baselines can be separated into three groups roughly as shown in Table 2.

The baselines in the first group are conducted on $\mathcal{D}_{17}$ dataset, covering the *AE*, *OE*, *ALSC*, and *AESC* subtasks. Span-based method SPAN-BERT (Hu et al., 2019) and sequence tagging method, IMN-BERT (He et al., 2019) and RACL-BERT (Chen and Qian, 2020), are selected. Specifically, the IMN-BERT model is reproduced by Chen and Qian (2020). All these baselines are implemented on BERT-Large.

The baselines of the second group are conducted on $\mathcal{D}_{19}$ dataset, mainly focusing on *AOE* subtask. Interestingly, we find that sequence tagging method is the main solution for this subtask (Fan et al., 2019; Wu et al., 2020; Pouran Ben Veyseh et al., 2020).

The baselines of the third group are mainly conducted on $\mathcal{D}_{20a}$ and $\mathcal{D}_{20b}$ datasets, which could cover almost all the ABSA subtasks except for one certain subtask depending on the baseline structures. For the following baselines: RINANTE (Dai and Song, 2019), CMLA (Wang et al., 2017), Li-unified (Li et al., 2019), the suffix “+” in Table 2 denotes the corresponding model variant modified by Peng et al. (2020) for being capable of *AESC*, *Pair* and *Triplet*.

4.3 Implement Details

Following previous studies, we use different metrics according to different subtasks and datasets. Specifically, for the single output subtasks *AE*, *OE*, and *AOE*, the prediction span would be considered as correct only if it exactly matches the start and the end boundaries. For the ALSC subtask, we require the generated sentiment polarity of the given aspect should be the same as the ground truth. As for compound output subtasks, *AESC*, *Pair* and *Triplet*, a prediction result is correct only when all the span boundaries and the generated sentiment polarity are accurately identified. We report the precision (P), recall (R), and F1 scores for all experiments[6].

4.4 Main Results

On $\mathcal{D}_{17}$ dataset (Wang et al., 2017), we compare our method for *AE*, *OE*, *ALSC*, and *AESC*. The comparison results are shown in Table 3. Most of our results achieve better or comparable results to

[6] Due to the limited space, we would present detailed experiments for each dataset in the Supplementary Material.

Model	14res				14lap				15res			
	AE	*OE*	*ALSC*	*AESC*	*AE*	*OE*	*ALSC*	*AESC*	*AE*	*OE*	*ALSC*	*AESC*
SPAN-BERT	86.71	-	71.75	73.68	82.34	-	62.5	61.25	74.63	-	50.28	62.29
IMN-BERT	84.06	85.10	75.67	70.72	77.55	**81.0**	75.56	61.73	69.90	73.29	70.10	60.22
RACL-BERT	86.38	87.18	81.61	75.42	81.79	79.72	73.91	63.40	73.99	76.0	**74.91**	66.05
Dual-MRC	86.60	-	**82.04**	**75.95**	82.51	-	75.97	65.94	75.08	-	73.59	65.08
Ours	**87.07**	**87.29**	75.56	73.56	**83.52**	77.86	**76.76**	**67.37**	**75.48**	**76.49**	73.91	**66.61**

Table 3: Comparison F1 scores for *AE*, *OE*, *SC*, and *AESC* on the $\mathcal{D}_{17}$ dataset (Wang et al., 2017). The baseline results are retrieved from Mao et al. (2021). We highlight the best results in bold. It is worth noting that all the baseline results are obtained via BERT-Large, while our results are obtained via BART-Base.

Model	14res			14lap			15res			16res		
	P	R	F1	P	R	F1	P	R	F1	P	R	F1
IOG	82.38	78.25	80.23	73.43	68.74	70.99	72.19	71.76	71.91	84.36	79.08	81.60
LOTN	84.0	80.52	82.21	77.08	67.62	72.02	76.61	70.29	73.29	86.57	80.89	83.62
ONG	83.23	81.46	82.33	73.87	77.78	75.77	76.63	**81.14**	78.81	87.72	84.38	86.01
Dual-MRC	**89.79**	78.43	83.73	78.21	**81.66**	79.90	77.19	71.98	74.50	86.07	80.77	83.33
Ours	86.01	**84.76**	**85.38**	**83.11**	78.13	**80.55**	**80.12**	80.93	**80.52**	**89.22**	**86.67**	**87.92**

Table 4: Comparison results for *AOE* on the $\mathcal{D}_{19}$ dataset (Fan et al., 2019). Baselines are from the original papers. We highlight the best results in bold.

Model	14res			14lap			15res			16res		
	AESC	*Pair*	*Triple.*	*AESC*	*Pair*	*Triple.*	*AESC*	*Pair*	*Triple.*	*AESC*	*Pair*	*Triple.*
CMLA+ †	70.62	48.95	43.12	56.90	44.10	32.90	53.60	44.60	35.90	61.20	50.00	41.60
RINANTE+ †	48.15	46.29	34.03	36.70	29.70	20.0	41.30	35.40	28.0	42.10	30.70	23.30
Li-unified+ †	73.79	55.34	51.68	63.38	52.56	42.47	64.95	56.85	46.69	70.20	53.75	44.51
Peng-two-stage †	74.19	56.10	51.89	62.34	53.85	43.50	65.79	56.23	46.79	71.73	60.04	53.62
JET-BERT ♯	-	-	63.92	-	-	50.0	-	-	54.67	-	-	62.98
Dual-MRC†	76.57	74.93	70.32	64.59	63.37	55.58	65.14	64.97	57.21	70.84	75.71	67.40
Ours	**78.47**	**77.68**	**72.46**	**68.17**	**66.11**	**57.59**	**69.95**	**67.98**	**60.11**	**75.69**	**77.38**	**69.98**

Table 5: Comparison F1 scores for *AESC*, *Pair* and *Triplet* on the $\mathcal{D}_{20a}$ dataset (Peng et al., 2020). The baseline results with "†" are retrieved from Mao et al. (2021), and result with "♯" is from Xu et al. (2020). We highlight the best results in bold.

Model	14res			14lap			15res			16res		
	P	R	F1	P	R	F1	P	R	F1	P	R	F1
CMLA+	39.18	47.13	42.79	30.09	36.92	33.16	34.56	39.84	37.01	41.34	42.1	41.72
RINANTE+	31.42	39.38	34.95	21.71	18.66	20.07	29.88	30.06	29.97	25.68	22.3	23.87
Li-unified+	41.04	**67.35**	51.0	40.56	44.28	42.34	44.72	51.39	47.82	37.33	54.51	44.31
Peng-two-stage	43.24	63.66	51.46	37.38	50.38	42.87	48.07	57.51	52.32	46.96	64.24	54.21
JET-BERT	**70.56**	55.94	62.40	55.39	47.33	51.04	**64.45**	51.96	57.53	**70.42**	58.37	63.83
Ours	65.52	64.99	**65.25**	**61.41**	**56.19**	**58.69**	59.14	**59.38**	**59.26**	66.6	**68.68**	**67.62**

Table 6: Comparison results for *Triplet* on the $\mathcal{D}_{20b}$ dataset (Xu et al., 2020). Baselines are from (Xu et al., 2020). We highlight the best results in bold.

baselines. However, these baselines yield competitive results based on the BERT-Large pre-trained models. While our results are achieved on the BART-Base model with almost half parameters. This shows that our framework is more suitable for these ABSA subtasks.

On $\mathcal{D}_{19}$ dataset (Fan et al., 2019), we compare our method for *AOE*. The comparison results are shown in Table 4. We can observe that our method achieves significant P/R/F1 improvements on 14res, 15res, and 16res. Additionally, we notice that our F1 score on 14lap is close to the previous SOTA result. This is probably caused by the dataset domain difference as the 14lap is the laptop comments while the others are restaurant comments.

On $\mathcal{D}_{20a}$ dataset (Peng et al., 2020), we compare our method for *AESC*, *Pair*, and *Triplet*. The comparison results are shown in Table 5. We can observe that our proposed method is able to outperform other baselines on all datasets. Specifically, we achieve the better results for *Triplet*, which demonstrates the effectiveness of our method on capturing interactions among aspect terms, opinion terms, and sentiment polarities. We also observe that the Span-based methods show superior performance to sequence tagging methods. This may be caused by the higher compositionality of candidate labels in sequence tagging methods (Hu et al., 2019). As the previous SOTA method, the Dual-MRC shows competitive performance by utilizing the span-based extraction method and the MRC mechanism. However, their inference process is not an end-to-end process.

On $\mathcal{D}_{20b}$ dataset (Xu et al., 2020), we compare our method for *Triplet*. The comparison results can be found in Table 6. Our method achieves the best results with nearly 7 F1 points improvements on 14res, 15res, and 16res. Our method achieves nearly 13, 9, 7, 12 points improvements on each dataset for the recall scores compared with other baselines. This also explains the drop performance of the precision score. Since $\mathcal{D}_{20b}$ is refined from $\mathcal{D}_{20a}$, we specifically compare the *Triplet* results of the corresponding dataset in $\mathcal{D}_{20a}$ and $\mathcal{D}_{20b}$. Interestingly, we discover that all baselines have a much bigger performance change on 15res. We conjecture the distribution differences may be the cause reason. In conclusion, all the experiment results confirm that our proposed method, which unifies the training and the inference to an end-to-end generative framework, provides a new SOTA solution for the whole ABSA task.

5 Framework Analysis

To better understand our proposed framework, we conduct analysis experiments on the $\mathcal{D}_{20b}$ dataset (Xu et al., 2020).

To validate whether our proposed framework could adapt to the generative ABSA task, we metric the invalid predictions for the *Triplet*. Specifically, since the *Triplet* requires the prediction format like $[a^s, a^e, o^s, o^e, s^p]$, it is mandatory that one valid triplet prediction should be in length 5, noted as "5-len", and obviously all end index should be larger than the corresponding start index, noted as "ordered prediction". We calculate $\frac{number\ of\ non{-}5{-}len}{total\ prediction}$, referred to as the "Invalid size", and the $\frac{number\ of\ non{-}ordered\ prediction}{total\ 5{-}len\ prediction}$, referred to as the "Invalid order". The "Invalid token" means the a^s is not the start of a token, instead, it is the index of an inside subword. From Table 7, we can observe that BART could learn this task form easily as the low rate for all the three metrics, which demonstrate that the generative framework for ABSA is not only a theoretically unified task form but also a realizable framework in practical. We remove these invalid predictions in our implementation of experiments.

As shown in Table 4, we give some analysis on the impact of the beam size, as we are a generation method. However, the beam size seems to have little impact on the F1 scores.

Errors	14res	14lap	15res	16res
Invalid size	0.48%	0.77%	1.41%	1.40%
Invalid order	1.75%	3.70%	3.26%	3.26%
Invalid token	0.48%	0.78%	1.02%	1.02%

Table 7: The errors for *Triplet* on the test set of the $\mathcal{D}_{20b}$.

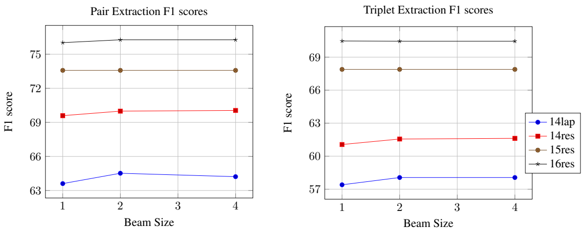


Figure 4: The F1 change curve with the increment of beam size on the dev set of $\mathcal{D}_{20b}$. The beam size seems to have little effect on the F1 scores.

6 Conclusion

This paper summarizes the seven ABSA subtasks and previous studies, which shows that there exist divergences on all the input, output, and task type sides. Previous studies have limitations on handling all these divergences in a unified framework. We propose to convert all the ABSA subtasks to a unified generative task. We implement the BART to generate the target sequence in an end-to-end process based on the unified task formulation. We conduct massive experiments on public datasets for seven ABSA subtasks and achieve significant improvements on most datasets. The experimental results demonstrate the effectiveness of our method. Our work leads to several promising directions, such as sequence-to-sequence framework on other tasks, and data augmentation.

Acknowledgements

We would like to thank the anonymous reviewers for their insightful comments. The discussion with colleagues in AWS Shanghai AI Lab was quite fruitful. We also thank the developers of fastNLP[7] and fitlog[8]. This work was supported by the National Key Research and Development Program of China (No. 2020AAA0106700) and National Natural Science Foundation of China (No. 62022027).

Ethical Considerations

For the consideration of ethical concerns, we would make detailed description as follows:

(1) All the experiments are conducted on existing datasets, which are derived from public scientific papers.

(2) We describe the characteristics of the datasets in a specific section. Our analysis is consistent with the results.

(3) Our work does not contain identity characteristics. It does not harm anyone.

(4) Our experiments do not need a lot of computer resources compared to pre-trained models.

(5) We will open source all our code.

References

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners.

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

Zhuang Chen and Tieyun Qian. 2020. Relation-aware collaborative learning for unified aspect-based sentiment analysis. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 3685–3694, Online. Association for Computational Linguistics.

Kyunghyun Cho, Bart van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using RNN encoder–decoder for statistical machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1724–1734, Doha, Qatar. Association for Computational Linguistics.

Hongliang Dai and Yangqiu Song. 2019. Neural aspect and opinion term extraction with mined rules as weak supervision. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5268–5277, Florence, Italy. 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.

Zhifang Fan, Zhen Wu, Xin-Yu Dai, Shujian Huang, and Jiajun Chen. 2019. Target-oriented opinion words extraction with target-fused neural sequence labeling. 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 2509–2518, Minneapolis, Minnesota. Association for Computational Linguistics.

Ruidan He, Wee Sun Lee, Hwee Tou Ng, and Daniel Dahlmeier. 2019. An interactive multi-task learning network for end-to-end aspect-based sentiment analysis. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 504–515, Florence, Italy. Association for Computational Linguistics.

Minghao Hu, Yuxing Peng, Zhen Huang, Dongsheng Li, and Yiwei Lv. 2019. Open-domain targeted sentiment analysis via span-based extraction and classification. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguis-*

[7] https://github.com/fastnlp/fastNLP. FastNLP is a natural language processing python package.

[8] https://github.com/fastnlp/fitlog. Fitlog is an experiment tracking package

tics, pages 537–546, Florence, Italy. Association for Computational Linguistics.

John D. Lafferty, Andrew McCallum, and Fernando C. N. Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the Eighteenth International Conference on Machine Learning (ICML 2001), Williams College, Williamstown, MA, USA, June 28 - July 1, 2001*, pages 282–289. Morgan Kaufmann.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: Denoising sequence-to-sequence pretraining 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.

Kun Li, Chengbo Chen, Xiaojun Quan, Qing Ling, and Yan Song. 2020. Conditional augmentation for aspect term extraction via masked sequence-to-sequence generation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7056–7066, Online. Association for Computational Linguistics.

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

Xin Li, Lidong Bing, Piji Li, and Wai Lam. 2019. A unified model for opinion target extraction and target sentiment prediction. In *The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI 2019, The Thirty-First Innovative Applications of Artificial Intelligence Conference, IAAI 2019, The Ninth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2019, Honolulu, Hawaii, USA, January 27 - February 1, 2019*, pages 6714–6721. AAAI Press.

Xin Li, Lidong Bing, Piji Li, Wai Lam, and Zhimou Yang. 2018b. Aspect term extraction with history attention and selective transformation. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI 2018, July 13-19, 2018, Stockholm, Sweden*, pages 4194–4200. ijcai.org.

Xin Li and Wai Lam. 2017. Deep multi-task learning for aspect term extraction with memory interaction. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2886–2892, Copenhagen, Denmark. Association for Computational Linguistics.

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

Thang Luong, Hieu Pham, and Christopher D. Manning. 2015. Effective approaches to attention-based neural machine translation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1412–1421, Lisbon, Portugal. Association for Computational Linguistics.

Dehong Ma, Sujian Li, and Houfeng Wang. 2018. Joint learning for targeted sentiment analysis. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4737–4742, Brussels, Belgium. Association for Computational Linguistics.

Dehong Ma, Sujian Li, Fangzhao Wu, Xing Xie, and Houfeng Wang. 2019. Exploring sequence-to-sequence learning in aspect term extraction. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3538–3547, Florence, Italy. Association for Computational Linguistics.

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

Yue Mao, Yi Shen, Chao Yu, and Longjun Cai. 2021. A joint training dual-mrc framework for aspect based sentiment analysis. *CoRR*, abs/2101.00816.

Margaret Mitchell, Jacqui Aguilar, Theresa Wilson, and Benjamin Van Durme. 2013. Open domain targeted sentiment. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1643–1654, Seattle, Washington, USA. 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 *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*, pages 8600–8607. AAAI Press.

Matthew Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages

2227–2237, New Orleans, Louisiana. Association for Computational Linguistics.

Maria Pontiki, Dimitris Galanis, John Pavlopoulos, Harris Papageorgiou, Ion Androutsopoulos, and Suresh Manandhar. 2014a. 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.

Maria Pontiki, Dimitris Galanis, John Pavlopoulos, Harris Papageorgiou, Ion Androutsopoulos, and Suresh Manandhar. 2014b. 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.

Maria Pontiki, Dimitris Galanis, John Pavlopoulos, Harris Papageorgiou, Ion Androutsopoulos, and Suresh Manandhar. 2014c. 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.

Amir Pouran Ben Veyseh, Nasim Nouri, Franck Dernoncourt, Dejing Dou, and Thien Huu Nguyen. 2020. Introducing syntactic structures into target opinion word extraction with deep learning. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 8947–8956, Online. Association for Computational Linguistics.

Xipeng Qiu, TianXiang Sun, Yige Xu, Yunfan Shao, Ning Dai, and Xuanjing Huang. 2020. Pre-trained models for natural language processing: A survey. *SCIENCE CHINA Technological Sciences*, 63(10):1872–1897.

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. *J. Mach. Learn. Res.*, 21:140:1–140:67.

Sunita Sarawagi and William W. Cohen. 2004. Semi-markov conditional random fields for information extraction. In *Advances in Neural Information Processing Systems 17 [Neural Information Processing Systems, NIPS 2004, December 13-18, 2004, Vancouver, British Columbia, Canada]*, pages 1185–1192.

Kaitao Song, Xu Tan, Tao Qin, Jianfeng Lu, and Tie-Yan Liu. 2019. MASS: masked sequence to sequence pre-training for language generation. In *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, volume 97 of *Proceedings of Machine Learning Research*, pages 5926–5936. PMLR.

Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. 2014. Sequence to sequence learning with neural networks. In *Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014, December 8-13 2014, Montreal, Quebec, Canada*, pages 3104–3112.

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

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

Yi Tay, Luu Anh Tuan, and Siu Cheung Hui. 2018. Learning to attend via word-aspect associative fusion for aspect-based sentiment analysis. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018*, pages 5956–5963. AAAI Press.

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

Oriol Vinyals, Meire Fortunato, and Navdeep Jaitly. 2015. Pointer networks. In *Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems 2015, December 7-12, 2015, Montreal, Quebec, Canada*, pages 2692–2700.

Wenya Wang and Sinno Jialin Pan. 2018. Recursive neural structural correspondence network for cross-domain aspect and opinion co-extraction. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2171–2181, Melbourne, Australia. Association for Computational Linguistics.

Wenya Wang, Sinno Jialin Pan, Daniel Dahlmeier, and Xiaokui Xiao. 2016a. Recursive neural conditional random fields for aspect-based sentiment analysis. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 616–626, Austin, Texas. Association for Computational Linguistics.

Wenya Wang, Sinno Jialin Pan, Daniel Dahlmeier, and Xiaokui Xiao. 2017. Coupled multi-layer attentions for co-extraction of aspect and opinion terms. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, February 4-9, 2017, San Francisco, California, USA*, pages 3316–3322. AAAI Press.

Yequan Wang, Minlie Huang, Xiaoyan Zhu, and Li Zhao. 2016b. 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.

Zhen Wu, Fei Zhao, Xin-Yu Dai, Shujian Huang, and Jiajun Chen. 2020. Latent opinions transfer network for target-oriented opinion words extraction. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*, pages 9298–9305. AAAI Press.

Hu Xu, Bing Liu, Lei Shu, and Philip S. Yu. 2018. Double embeddings and CNN-based sequence labeling for aspect extraction. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 592–598, Melbourne, Australia. Association for Computational Linguistics.

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

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

Meishan Zhang, Yue Zhang, and Duy-Tin Vo. 2015. Neural networks for open domain targeted sentiment. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 612–621, Lisbon, Portugal. Association for Computational Linguistics.

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

He Zhao, Longtao Huang, Rong Zhang, Quan Lu, and Hui Xue. 2020. SpanMlt: A span-based multi-task learning framework for pair-wise aspect and opinion terms extraction. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 3239–3248, Online. Association for Computational Linguistics.

A Supplemental Material

A.1 Experimental Environment

We use the triangular learning rate warmup. All experiments are conducted in the Nvidia Ge-Force RTX-3090 Graphical Card with 24G graphical memory.

The averages running time for experiments on each dataset is less than 15 minutes. The number of parameters is as follows:

• BART-Base model: 12 layers, 768 hidden dimensions and 16 heads with the total number of parameters, 139M;

• BERT-Base model: 12 layers, 768 hidden dimensions and 12 heads with the total number of parameters, 110M.

A.2 Decoding Algorithm for Different Datasets

In this part, we introduce the decoding algorithm we used to convert the predicted target sequence Y into the target span set L. These algorithm can be found in Algorithm 2, 3, 4.

Algorithm 2 Decoding Algorithm for the *AOE* subtask

```
Input: Number of tokens in the input sentence $n$, target sequence $Y = [y_1, ..., y_m]$ and $y_i \in [1, n + |C|]$, $L_T$ is a given length for different tasks.
Output: Target span set $L = \{(o_1^s, o_1^e, ..., o_i^s, o_i^e)\}$
$L = \{\}, e = [], i = 3$
while $i <= m$ do
    $y_i = Y[i]$
    $e.append(y_i)$
    $i += 1$
end while
$L.add(e)$
return $L$
```

Algorithm 3 Decoding Algorithm for the *AESC* Subtask

```
Input: Number of tokens in the input sentence $n$, target sequence $Y = [y_1, ..., y_m]$ and $y_i \in [1, n + |C|]$
Output: Target span set $L = \{(a_1^s, a_1^e, s_1), ..., (a_i^s, a_i^e, s_i)\}$
$L = \{\}, e = [], i = 1$
while $i <= m$ do
    $y_i = Y[i]$
    if $y_i > n$ then
        $L.add((e, C_{y_i - n}))$
        $e = []$
    else
        $e.append(y_i)$
    end if
    $i += 1$
end while
return $L$
```

Algorithm 4 Decoding Algorithm for the *AE/OE/Pair* subtasks

```
Input: Number of tokens in the input sentence $n$, target sequence $Y = [y_1, ..., y_m]$ and $y_i \in [1, n + |C|]$, $L_T$ is a given length for different tasks.
Output: Target span set $L = \{x_1, ..., x_i\}$($x_i$ is $(a_i^s, a_i^e)$, $(o_i^s, o_i^e)$ and $(a_i^s, a_i^e, o_i^s, o_i^e)$ for AE/OE/Pair, respectively)
$L = \{\}, e = [], i = 1$
while $i <= m$ do
    $y_i = Y[i]$
    if $len(e) == L_T$ then
        $L.add((e, C_{y_i - n}))$
        $e = []$
    end if
    $e.append(y_i)$
    $i += 1$
end while
return $L$
```

A.3 Detailed Experimental Setup

Experiments on each dataset

As the different subtasks are conducted on different datasets, specifically, we conduct the following experiments on each dataset:

• On the $\mathcal{D}_{17}$ dataset, we conduct the AESC and the OE in multi-task learning method. To that end, we feed the pre-defined task tags "<AESC>" and "<OE>" to the decoder first. For example, for the input "*The **drinks** are always well made and **wine selection** is fairly priced*" from $\mathcal{D}_{17}$ dataset, we define the AESC sequence and the OE target sequence as "<AESC>, 1, 1, POS, 7, 8, POS, </s>" and "<OE>, 4, 5, 10, 11, </s>".

• On the $\mathcal{D}_{19}$ dataset, we conduct the AOE. As the AOE subtask requires to detect the opinion terms given aspect terms in advance, the aspect terms need to be fed to our decoder first. For the aforementioned example sentence from $\mathcal{D}_{19}$ dataset, we define the AOE target sequence as " 1, 1, 4, 5, </s>" and the " 7, 8, 10, 11, </s>".

• On the $\mathcal{D}_{20a}$ and $\mathcal{D}_{20b}$ datasets, we conduct the Triplet Extraction. For the aforementioned example sentence from $\mathcal{D}_{20a}$ and $\mathcal{D}_{20b}$ dataset, we define the Triplet target sequence as "1, 1, 4, 5, POS, 7, 8, 10, 11, POS, </s>".

Specific Subtask Metrics

• On the $\mathcal{D}_{17}$ dataset, we get the AESC and OE results directly. Following previous work, we only calculate the metrics for AESC and ALSC from those true positive AE predictions. Specifically, the F1

• On the $\mathcal{D}_{19}$ dataset, we get the AOE results directly. The metrics for AOE are standard Precision, Recall and the F1 score.

• On the $\mathcal{D}_{20a}$ and $\mathcal{D}_{20b}$ datasets, we get the Triplet results directly. We preserve the <AT,OT> for Pair metric and <AT, SP> for AESC metric. The metrics for them are standard Precision, Recall and the F1 score.