

# LEGO-ABSA: A Prompt-based Task Assemblable Unified Generative Framework for Multi-task Aspect-based Sentiment Analysis

Tianhao Gao<sup>1,\*</sup>, Jun Fang<sup>2,\*</sup>, HanyuLiu<sup>2,\*</sup>, Zhiyuan Liu<sup>2</sup>  
Chao Liu<sup>2,†</sup>, Pengzhang Liu<sup>2</sup>, Yongjun Bao<sup>2</sup>, Weipeng Yan<sup>2</sup>  
<sup>1</sup> School of Software and Microelectronics, Peking University  
<sup>2</sup> JD, Retail, Beijing, China  
gaotianhao@pku.edu.cn  
{junfang8, liuhanyu11, liuzhiyuan8,  
liuchao397, liupengzhang, baoyongjun, Paul.yan}@jd.com

## Abstract

Aspect-based sentiment analysis (ABSA) has received increasing attention recently. ABSA can be divided into multiple tasks according to the different extracted elements. Existing generative methods usually treat the output as a whole string rather than the combination of different elements and only focus on a single task at once. This paper proposes a unified generative multi-task framework that can solve multiple ABSA tasks by controlling the type of task prompts consisting of multiple element prompts. Further, the proposed approach can train on simple tasks and transfer to difficult tasks by assembling task prompts, like assembling Lego bricks. We conduct experiments on six ABSA tasks across multiple benchmarks. Our proposed multi-task approach achieves new state-of-the-art results in almost all tasks and competitive results in task transfer scenarios.

## 1 Introduction

ABSA is a fine-grained sentiment analysis task that has attracted increasing attention in recent years (Schouten and Frasincar, 2016; Nazir et al., 2020). ABSA aims to extract different elements including: 1) the aspect term ( $a$ ); 2) opinion term ( $o$ ); 3) the aspect category ( $c$ ) corresponding to the aspect term; 4) the sentiment polarity ( $s$ ) for a specific aspect term. For example, in the sentence “Pizza is delicious”, “Pizza” is an aspect term belonging to the food category, and the corresponding opinion term is “delicious”, which expresses positive sentiment. As shown in Table 1, based on the combination of different elements to be extracted, ABSA can be divided into multiple tasks.

This paper explores tasks containing two or more elements. In general, most ABSA tasks are transferred to classification tasks. Previous works often

designed a new architecture carefully and trained with the corresponding dataset for a specific sub-task. We review them as follows:

**Pair Extraction** The pair extraction task includes AOPE (Aspect-Opinion Pair Extraction), ACSA (Aspect-Category Sentiment Analysis) and E2E-ABSA (End-to-End Aspect-based Sentiment Analysis) in our method. ACSA is usually treated as a multi-task classification task (Hu et al., 2018; Dai et al., 2020; Ma et al., 2018). Some works convert the AOPE and E2E-ABSA tasks into sequence tagging problems (Wu et al., 2020b; Gao et al., 2021; Chen et al., 2020; He et al., 2019), specifically using the BIO tagging strategies (Wang and Pan, 2018; Wu et al., 2020a; Li et al., 2019a,b). The pair extraction is also named the basic task in this paper.

**Triples Extraction** The triples extraction tasks contain ASTE (Aspect Sentiment Triplet Extraction) and TASD (Target Aspect Sentiment Detection) in our paper. Most previous works still treat them as a sequence tagging task (Xu et al., 2020, 2021; Zhang et al., 2020; Wu et al., 2021). Some works transfer them to Machine Reading Comprehension tasks (Mao et al., 2021; Chen et al., 2021).

**Quadruple Extraction** (Cai et al., 2021) firstly introduced the quadruple extraction task, i.e., ASQP (Aspect Sentiment Quad Prediction), and provide a multi-stage classification structure adopting from an aspect-opinion co-extraction system (Wang et al., 2017).

Recently, large-scale generative language models have become increasingly powerful (Raffel et al., 2020; Lewis et al., 2019; Radford et al., 2019), and any ABSA task can be converted to a generative problem. Some generative frameworks (Zhang et al., 2021b,a; Yan et al., 2021; Hosseini-Asl et al., 2022) have been proposed and achieved state-of-the-art results in the field of ABSA. The genera-

\*These authors have equal contribution. Work done during gaotianhao’s internship at JD, Retail, Beijing, China

†Corresponding authors.

Task Name	Input	Output
Aspect-Opinion Pair Extraction(AOPE)	Pizza is delicious	Pizza, delicious (a,o)
Aspect-Category Sentiment Analysis(ACSA)		food, delicious (c,s)
End-to-End Aspect-based Sentiment Analysis (E2E-ABSA)		Pizza, Positive (a,s)
Aspect Sentiment Triplet Extraction(ASTE)		Pizza, delicious, positive (a,o,s)
Target Aspect Sentiment Detection(TASD)		Pizza, food, positive(a,c,s)
Aspect Sentiment Quad Prediction(ASQP)	Pizza, delicious, food, positive (a,o,c,s)	

Table 1: target of different tasks

tive format includes but is not limited to Generating Structure-Linearized Texts, Labelaugmented Text(Zhang et al., 2021b) Generating Word Indices (Yan et al., 2021) Filling Templates (Zhang et al., 2021a), as summarized by (Min et al., 2021).

However, all generative approaches mentioned above suffer from 1) training and predicting a single specific task at once; 2) treating output as a whole text rather than a combination of individual elements; 3) poor transferability from simple task to difficult task. Below is a detailed description of these three points.

For the first point, in mentioned generative approaches, the input and output formats do not support training on multiple ABSA tasks simultaneously, which we call **multi-task training setting**.

For the second point, in previous works, the models cannot understand the meaning of each element to be extracted because the input and output are treated as simple strings, and the model completes the task of predicting output through auto-regression.

For the third point, previous methods cannot be applied to task transfer scenarios. Compared to the triplets like ASTE, pairs like AOPE and E2E-ABSA are much easier in the annotation. However, previous works cannot complete ASTE by training only on AOPE and E2E-ABSA tasks even though the ASTE task elements are the same as the union of AOPE and E2E-ABSA task elements. We call this **task transfer scenario**, and it is a special case under a multi-task training setting. The proposed method has a competitive performance in this setting.

Inspired by above observations, we propose a unified generative framework LEGO-ABSA that can simultaneously solve multiple ABSA tasks and transfer from simple to complex tasks. Specifically, we take T5 as our backbone network and combine prompt learning with the practice of placing sentinel tokens of T5 pre-training. Unlike most previous works that use a piece of simple text as a prompt, e.g."ASQP" in (Zhang et al., 2021a), we

design an element prompt and establish the correspondence between each element with the element prompt. We make the framework treat prompt and output text as a combination of independent elements by this design. We combine multiple element prompts into task prompt. The task prompt of a simple task can be regarded as basic bricks which can be assembled to transfer to a complex task, just like assembling Lego bricks. The output sequence is formed as a concatenation of the sentinel tokens and the real answer tokens, consistent with T5. To verify the effectiveness of our method, we conduct experiments on public datasets. Comparison results show that our proposed framework outperforms previous state-of-the-art (SOTA) approaches in most tasks. Moreover, in the case of missing part of the data annotation, it can also achieve competitive performance.

In summary, our main contributions are as follows:

- We propose a prompt-based unified generative framework to solve all ABSA tasks. The framework can be trained on multiple tasks simultaneously, and it also performs competitively in task transfer scenarios.
- To the best of our knowledge, we are the first to explore solutions for task transfer scenarios.
- The experimental results show that Our method significantly outperforms the SOTA methods on E2E-ABSA, AOPE, ASTE, and ACSA tasks.

## 2 Methodology

### 2.1 Task formulation

The proposed method will formulate any ABSA task as a text generation task. Here we give formal definitions of generative frameworks' inputs and output text.

The input  $x$  consists of two part, the raw text  $t$  and a **task prompt**  $p$ :  $x = t + | + p_{task}$ .  $t = [t_1, t_2, \dots, t_n]$  where  $t_i$  is the  $i$ th token of  $t$  and  $n$  is

the length of tokens.  $p_{task} = [p_1, p_2, \dots, p_{m_{task}}]$  where  $p_i$  is the  $i$ th element prompt of  $p_{task}$  and  $m_{task}$  is the number of element prompt in  $p_{task}$ , which is used as a condition to generate different output text for different task.

Output text  $o_{task} = [o_1, o_2, \dots, o_{m'}]$ , where  $o_i$  is the  $i$ th tokens pair of  $o_{task}$  and  $m'$  is the output length based on the current input  $x$ . The subsequent subsection will describe construction methods in detail.

## 2.2 Element Prompt Definition

### 2.2.1 Introduction of T5

T5 is an encoder-decoder model pre-trained on a multi-task mixture of unsupervised and supervised tasks converted into a text-to-text format.

In order to minimize the gap in pre-training and fine-tune, we use the same training mode as the T5 dose in pre-training. The goal of T5 is similar to the cloze test. As shown in the Figure 1(a), the input of T5 is a sentence with randomly masked consecutive spans using sentinel tokens. During unsupervised training, T5 aims to reconstruct the continuous span masked by the sentinel token, i.e.,  $\langle \text{extra\_id\_i} \rangle$  in the Figure 1(a) incrementing one by one starting from zero. Through this training object, T5 can learn general language features.

### 2.2.2 Element Prompt

In order to make the framework fully understand the meaning of each element in the output text, instead of treating the output as a simple string, we design an element prompt for each extracted element.

We define the element prompt as "aspect:  $\langle \text{extra\_id\_0} \rangle$ ", which has two advantages. On the one hand, the format is consistent with the T5 unsupervised training object, which can help us make better use of the information learned from pre-training. On the other hand, by defining a prompt for a single element, the output is no longer regarded as a whole text string but as a combination of different elements that offer more convenience.

The element prompts for the four elements in the ABSA task are as follows. We use w, x, y, and z to represent the id of the sentinel token.

- $p_a$  : "aspect :  $\langle \text{extra\_id\_w} \rangle$ "
- $p_c$  : "category :  $\langle \text{extra\_id\_x} \rangle$ "
- $p_o$  : "opinion :  $\langle \text{extra\_id\_y} \rangle$ "
- $p_s$  : "sentiment :  $\langle \text{extra\_id\_z} \rangle$ "

## 2.3 Task Prompt of Single-task Training

From shallow to deep, we start with the single ABSA task.

The element prompt is defined for each element to be extracted, but in order to complete a specific ABSA task, we need to concatenate different element prompts to form the task prompt, i.e.,  $p_{task}$ .  $p_{task}$  is used as a condition so that the backbone can distinguish between different tasks. According to the kind of elements extracted and the order of element extraction in each task, we concatenate all element prompts by commas, e.g.  $p_{AOPE}$  can be  $p_{ao}$  or  $p_{oa}$  which means  $p_a+$ ,  $+p_o$  and  $p_o+$ ,  $+p_a$ . Because the training for each task is independent, it is trivial to maintain a unique mapping relationship between sentinel token id and element. Here, sentinel token id for each task increments from 0, as shown in the Figure 1(b) with the sample of AOPE. The rest of the task prompts also follow the same method to define.

The arrangement order of the element prompt matters since the generation model is generated in an auto-regressive manner, and the elements generated first can provide more prior information for the elements generated later. From our experimental observations, the elements are arranged in priority according to **aspect term > opinion term = aspect category > sentiment polarity**

## 2.4 Task Prompt of Multi-task Training

An improvement of our framework is the ability to organize multiple ABSA tasks into a multi-task training task through task prompts.

As shown in Figure 1(c), under the multi-task training setting, the task prompt is still constructed by concatenating element prompt like the single task. The difference is that the one-to-one correspondence between elements and sentinel tokens is shared between multiple sub-tasks, so we define a global mapping relationship between the sentinel token and the corresponding element. Following the priority of elements mentioned above we assign  $\langle \text{extra\_id\_0} \rangle$  to aspect term,  $\langle \text{extra\_id\_1} \rangle$  to opinion term,  $\langle \text{extra\_id\_2} \rangle$  to aspect category and  $\langle \text{extra\_id\_3} \rangle$  to sentiment polarity. After setting each task prompt, we concatenate task prompts to each original input of the corresponding task and then mix the data of all tasks to do multi-task training.

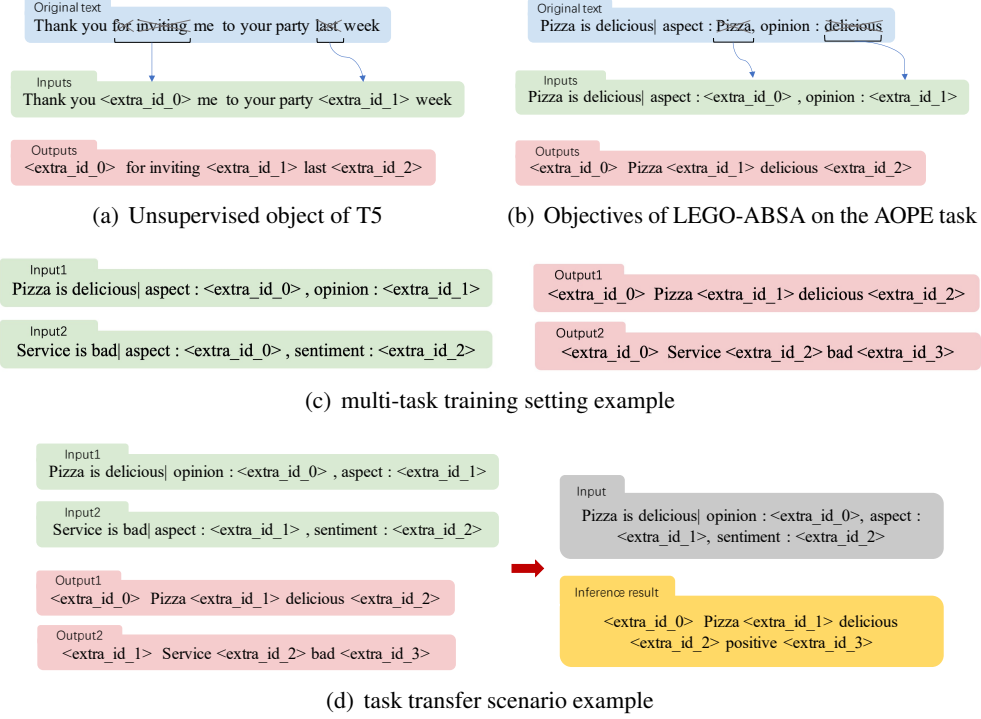


Figure 1

### 2.4.1 Task Transfer Scenario

This section introduces how the proposed framework works in a task transfer scenario. As shown in Figure 1(d), we define the task that extracts two elements and the combination relationship between elements as **basic task**. As illustrated in Figure 2, AOPE, E2E-ABSA, and ACSA are **basic tasks** to accomplish more complicated tasks. Basic tasks can be regarded as the bricks in LEGO.

We call the overlapping element of any two basic tasks **connection element** which is like a connector that connects two bricks. We define ASTE, TASD, and ASQP as **advanced task** which aims to extract three or more elements and the combination relationship between elements. The advanced task is like a final product assembled from basic bricks and connectors. **The goal for task transfer scenario is to resolve advanced tasks only given the training data of basic tasks**, and the process of using the basic tasks to construct the advanced tasks is like building Lego.

To achieve this goal, we need to figure out two questions: what basic tasks are required for a given advanced task; how to assemble the basic tasks, i.e., the way to construct an advanced task prompt from basic task prompts. We will give a detailed introduction in the following section for these two questions.

**Basic Task Confirmation** In order to complete the advanced task, we need to confirm the corresponding basic task. Because advanced tasks consist of basic tasks connected by connection elements, we need two basic tasks for extraction of triplet, like ASTE and TASD. For ASQP extraction, we need all three basic tasks mentioned in this paper. Then according to the elements contained in the task, we can determine that the basic tasks of ASTE(element set is  $\{o, a, s\}$ ) are AOPE(element set is  $\{o, a\}$ ) and E2E-ABSA(element set is  $\{a, s\}$ ). The basic tasks of TASD(element set is  $\{a, c, s\}$ ) are E2E-ABSA(element set is  $\{a, s\}$ ) and ACSA(element set is  $\{c, s\}$ ). The basic tasks of ASQP(element set is  $\{a, o, c, s\}$ ) are AOPE(element set is  $\{o, a\}$ ), E2E-ABSA(element set is  $\{a, s\}$ ) and ACSA(element set is  $\{c, s\}$ ).

**Task Prompt Assemble** After confirming the basic task, we will illustrate the method of task prompt assemble, and the arrangement order of elements is essential here. We denote the advanced task as  $A$  and its basic tasks set as  $\mathbb{B}$ .

We initialize  $p_A = \text{""}$ , then take any element that is not a connection element as the start element and let  $p_A = p_A + \text{","} + p_{start}$ . Next, we select a task  $B$  containing start element from  $\mathbb{B}$  and use another element in  $B$  as the next element, then delete



$B$  from  $\mathbb{B}$ . Afterward, we use the next element as the start element and repeat the above process until all elements of  $A$  have been traversed once.

For example, as shown in Figure 2, given ASQP as advanced task, we can get  $\mathbb{B} = \{\{a, s\}, \{a, o\}, \{c, s\}\}$ , and the connection elements are  $\{a, s\}$ . First, we choose element  $o$  as the beginning element and the corresponding task  $B$  is  $\{a, o\}$ . Then, we concatenate  $p_o$  with  $p_A$  and let another element  $a$  be the new begin element and delete  $\{a, o\}$  from  $\mathbb{B}$ . Next, the corresponding task  $B$  is  $\{a, s\}$  based on element  $a$ . Afterward, we repeat the process until all elements of ASQP have been traversed once. Through the above process, the result is shown in Figure 2,  $p_A = p_o + \text{“,”} + p_a + \text{“,”} + p_s + \text{“,”} + p_c$ .

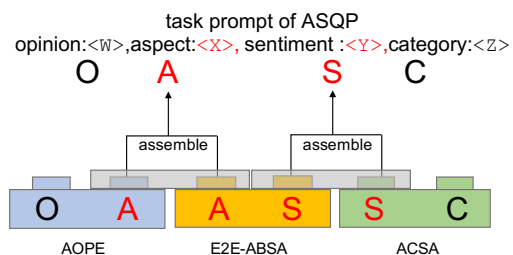


Figure 2: Illustration of task prompt assemble

We can get the task prompt after assembly and determine the order of element prompts through the above process. Besides, we also need to maintain the global mapping relationship between the sentinel token and element. As shown in Table 2, we design a set of task prompts that are unique in global mapping and conform to assemble rules. Moreover, the priority order of elements of the task prompt in Table 2 is preserved as much as possible.

## 2.5 Output Sequence Definition

As shown in Figure 1(b), the definition form of the  $o_{task}$  is consistent with the T5 unsupervised training output form. Whether it is a multi-task training or task transfer setting, the output sequence is formed as a concatenation of the sentinel tokens and the corresponding gold label. For sentences with multiple sets of extracted elements, we use “,” to separate sets.

## 3 Experiment

### 3.1 Datasets

We evaluate the proposed LEGO-ABSA on benchmark SemEval14-16 initially provided by the SemEval shared challenges (Pontiki et al., 2014, 2015,

2016). For each ABSA task, we use the public datasets derived from SemEval14-16 with additional sentiment annotations. Specifically, we adopt the dataset AOPE, ASTE, and E2E-ABSA provided by (Peng et al., 2020), ACSA provided by (Pontiki et al., 2015, 2016; Liu et al., 2021) TASD provided by (Wan et al., 2020), ASQP provided by (Zhang et al., 2021a). For a fair comparison, we use the same data split as previous works.

### 3.2 Baselines

For E2E-ABSA, AOPE, and ASTE tasks, we adopt two types of baselines: 1) extraction based methods, including Li-unified(Li et al., 2019a), Peng-two-stage(Peng et al., 2020) and Bi-MRC(Chen et al., 2021) JET-BERT(Xu et al., 2020), Dual-MRC(Mao et al., 2021); 2) generation based methods, including GAS(Zhang et al., 2021b) and Yan-unified(Yan et al., 2021)

For the ACSA task, we adopt five baselines derived from (Cai et al., 2020). For TASD and ASQP tasks, we utilize two types of baselines 1) Extraction-based methods, including TAS-LPM-CRF and TAS-SW-TO from (Wan et al., 2020) and TASO-BERT-CRF (Zhang et al., 2021a); 2) Generation-based methods including GAS(Zhang et al., 2021b) and PARAPHRASE(Zhang et al., 2021a).

### 3.3 Implementation Details

**Evaluation Metrics** F1 score is the evaluation metric for all tasks. A prediction is correct if all its predicted sentiment elements in the pair, triplet, or quadruple are correct.

**Experiment Details** We adopt the pre-trained T5-base model released by *huggingface*\*. We set the learning rate to  $3e-4$  as suggested by *huggingface*. In single task and multi-task training settings, the model is trained up to 20 epochs for the AOPE, E2E-ABSA, ACSA, and ASTE tasks and 30 epochs for the TASD and ASQP tasks. We train two multi-task models according to whether the aspect category element is included. The first is trained with AOPE, E2E-ABSA, and ASTE tasks, while the second model is trained with ACSA, TASD, and ASQP. For the in-domain task transfer setting, we train one epoch on basic tasks and 2 epochs on basic tasks with a learning rate of  $3e-4$ . For the cross-domain setting, we train five epochs on basic tasks with a learning rate equal to  $3e-4$ .

\*<https://huggingface.co/t5-base>

Task name	Task prompt
AOPE	opinion:<extra_id_0>, aspect:<extra_id_1>
E2E-ABSA	aspect:<extra_id_1>, sentiment:<extra_id_2>
ACSA	sentiment:<extra_id_2>, category:<extra_id_3>
ASTE	opinion:<extra_id_0>, aspect:<extra_id_1>, sentiment:<extra_id_2>
TASD	aspect:<extra_id_1>, sentiment:<extra_id_2>, category:<extra_id_3>
ASQP	opinion:<extra_id_0>, aspect:<extra_id_1>, sentiment:<extra_id_2>, category:<extra_id_3>

Table 2: Task prompts in task transfer scenarios

### 3.4 Main Results

The main results show in Table 3 and 4. All results are the average F1 scores across 3 runs with different random seeds.

Notably, our proposed method with a single task outperforms the state-of-the-art on AOPE, E2E-ABSA, ASTE, and ACSA tasks by 1.9, 2.4, 1.7, and 3.1 average F1 scores, respectively. Besides, competitive results are also shown on TASD and ASQP.

Our method with a multi-task training setting achieved more competitive performance than separate training for each task, even though we only used one T5-base as the backbone. We get 2.5, 3.3, 2.5, and 1.6 average higher F1 scores than the state-of-the-art methods on AOPE, E2E-ABSA, ASTE, and ACSA tasks. For AOPE, E2E-ABSA, and ASTE tasks. Our model is trained on four datasets on each task and only uses one backbone, which is equivalent to reducing the backbone size to 1/12 compared with the previous method with one model per task, while the average F1 is 2.8 points higher. The result shows that multi-task training can significantly improve performance. Since the multi-task training is modeled under a unified generative framework, the construction of input and output follows the same principle so that the information between different tasks can be utilized and mutually enhanced.

Regarding why TASD and ASQP do not perform as outstanding as the rest of the tasks, we speculate that it may be because TASD and ASQP both need to extract aspect category and sentiment polarity. These two elements are generated by reasoning and have not appeared in the original text. The unsupervised pre-training object of T5 can only guarantee to generate text spans that have appeared in the original text. The working principle of sentiment and category extraction is similar to using a generative model to do classification tasks, which is different from the unsupervised training object of T5. The gap between tasks is the leading cause of performance degradation.

### 3.5 Task Transfer Results

This section verifies our proposed framework’s in-domain and cross-domain performance under the task transfer scenario.

#### 3.5.1 In-domain

In the in-domain setting, we complete a advanced task by training on the necessary basic tasks of the same training corpus at a time. The result of the in-domain task transfer is shown in Table 5. We were surprised to find that the inference performance on advanced tasks is very competitive by training on basic tasks. Even the result on the ASTE task surpasses some purely supervised baselines.

#### 3.5.2 Cross-domain

In some real situations, AOPE and E2E-ABSA annotations may not be on the same corpus, or we cannot combine them into a complete ASTE annotation. Therefore, task transfer performance of cross-domain is very important.

For TASD and ASQP tasks, since the cross-domain aspect categories are not the same, the model cannot transfer across domains on tasks that include aspect categories. Therefore, we conduct experiments on ASTE in this section under the cross-domain setting.

The cross-domain result shows in Table 6. The proposed method outperforms some purely supervised methods on average, and no noticeable performance drop compared to the in-domain setting. Compared with rule-based methods, task prompt assembly can achieve a large performance improvement. The possible reason is that, in the rule-based approach, the error of each model caused by domain transfer propagates. However, the task prompt assembly is more similar to a joint method. Therefore, the performance promotion is obvious.

## 4 Analysis

This section explores the principle of T5 assembly basic task corresponding to task prompt under task transfer training setting.

Model	AOPE				E2E-ABSA				ASTE			
	L14	R14	R15	R16	L14	R14	R15	R16	L14	R14	R15	R16
Li-unified(Li et al., 2019a)	52.6	55.3	56.9	53.8	63.4	73.8	65.0	70.2	42.5	51.7	46.7	44.5
Peng-two-stage(Peng et al., 2020)	53.9	56.1	56.2	60.0	62.3	74.2	65.8	71.7	43.5	51.9	46.8	53.6
JET-BERT(Xu et al., 2020)	-	-	-	-	-	-	-	-	50.0	63.9	54.7	62.9
Bi-MRC(Chen et al., 2021)	67.4	76.2	68.6	76.5	67.2	76.3	67.1	73.1	59.2	70.6	61.0	68.1
Dual-MRC(Mao et al., 2021)	63.3	74.9	64.9	75.7	64.5	76.5	65.1	70.8	55.5	70.3	57.2	67.4
GAS(Zhang et al., 2021b)	63.8	73.2	65.0	75.0	65.3	78.5	69.4	72.7	54.5	70.2	59.1	65.0
Yan-unified(Yan et al., 2021)	66.1	77.7	68.0	<b>77.4</b>	68.2	78.5	70.0	75.7	57.6	72.5	60.1	70.0
LEGO-ABSA(multi-task)	<b>71.3</b>	<b>78.0</b>	<b>72.9</b>	77.1	<b>72.3</b>	<b>80.6</b>	<b>74.2</b>	76.1	<b>62.2</b>	<b>73.7</b>	<b>64.4</b>	69.9
LEGO-ABSA(separate)	69.7	<b>78.1</b>	71.4	<b>77.6</b>	69.1	80.0	<b>74.3</b>	<b>78.6</b>	59.5	72.6	63.2	<b>71.5</b>

Table 3: Main result on AOPE, E2E-ABSA, and ASTE tasks. LEGO-ABSA(multi-task) means mixing the training dataset of three tasks and shuffling the order. LEGO-ABSA(separate) means that a task is trained with only one dataset, like other baselines. Since the original paper of GAS is not implemented on Peng’s dataset, we reproduce the results ourselves using the same experiment config. We highlight the best results and results with F1 gaps within 0.2

Model	ACSA				Model	TASD		ASQP	
	L15	L16	R15	R16		R15	R16	R15	R16
Cartesian-BERT	32.8	39.5	58.4	68.9	TAS-SW-TO(Wan et al., 2020)	58.1	65.4	-	-
AddOneDim-BERT	48.9	47.2	61.7	69.8	TAS-LPM-CRF(Wan et al., 2020)	54.7	64.6	-	-
Hier-BERT	50.6	49.2	62.4	70.3	TASO-BERT-CRF(Zhang et al., 2021a)	-	-	34.8	43.7
Hier-Transformer-BERT	57.8	52.7	64.7	73.5	GAS(Zhang et al., 2021b)	60.6	68.3	46.0	56.0
Hier-GCN-BERT	62.13	54.2	64.2	74.6	PARAPHRASE(Zhang et al., 2021a)	<b>63.1</b>	<b>72.0</b>	<b>46.9</b>	<b>57.9</b>
LEGO-ABSA(multi-task)	<b>65.0</b>	53.6	67.3	75.6	LEGO-ABSA(multi-task)	62.3	<b>71.8</b>	46.1	57.6
LEGO-ABSA(separate)	64.2	<b>55.9</b>	<b>71.0</b>	<b>76.2</b>	LEGO-ABSA(separate)	61.7	68.8	45.8	<b>57.7</b>

Table 4: Main results on ACSA, TASD, and ASQP tasks. LEGO-ABSA(multi-task) means mixing individual training dataset and shuffling the order. We highlight the best results and results with F1 gaps within 0.2 in bold.

Task	L14	R14	R15	R16	GM	TTO	TST	task transfer ability
ASTE	49.2	60.9	51.4	50.0	×	×	✓	×
TASD	-	-	30.9	30.6	✓	×	✓	×
ASQP	-	-	25.8	24.5	✓	✓	×	×
					✓	✓	✓	✓

Table 5: In-domain task transfer performance. In this situation, basic tasks and advanced task are on the same domain and corpus.

method	Lap → Rest	Rest → Lap
GAS-rule	32.4	33.7
LEGO-ABSA	<b>53.9</b>	<b>44.7</b>

Table 6: Cross-domain task transfer performance on ASTE task. We use dataset from Peng(Peng et al., 2020). Where rule method means that we get results by combining  $(a, s)$  and  $(a, o)$  with same  $a$ .

#### 4.1 Factor Analysis of Transferability

We try to 1) increase the sentinel token id from 0 in each basic task, which means no global mapping between sentinel token and element. 2) give a global sentinel token id for each element but randomly arrange elements’ order in the basic task. 3) employ the global mapping and right order of

Table 7: Factor analysis for task transferability, where GM is global mapping between sentinel token and element, TTO is task transfer order that follows rule of Task Prompt Assemble 2.4.1, and TST is use original T5 sentinel token instead of custom token.

element prompts, but replace  $\langle \text{extra\_id\_x} \rangle$  (the sentinel token used in T5 pre-training) with a custom new token. As shown in Table 7, only when all three conditions are met can the backbone obtain the task transferability.

Using the T5 Sentinel Token shows that downstream tasks can indeed reuse the unsupervised output of T5 pre-training. The custom token cannot have the function of masking a consecutive span because it has not been pre-trained. The global mapping between sentinel token and element shows that each sentinel token has a specific meaning after downstream task training. More importantly, the experiment result shows that a specific element

task prompt	Prediction
aspect: <extra_id_0>	tech support
opinion: <extra_id_1>	not fix
sentiment: <extra_id_2>	negative
aspect: <extra_id_0>, opinion: <extra_id_1>	tech support, not fix
aspect: <extra_id_0>, sentiment: <extra_id_2>	tech support, negative
opinion: <extra_id_1>, sentiment: <extra_id_2>	not fix, negative

Table 8: Lego split case for text "tech support would not fix the problem unless I bought your plan for \$ 150 plus ."

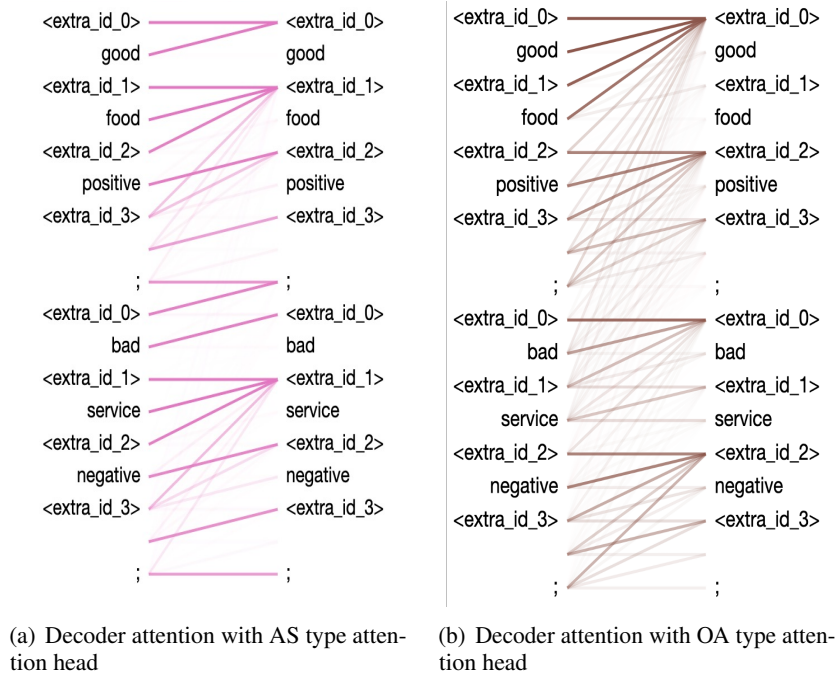


Figure 3: attention visualization

prompt arrangement must be used to achieve task transfer and indirectly show that what the backbone learns is how to mix two or more task prompts.

### Decoder Attention Visualization

We conjecture that LEGO-ABSA uses the ending element prompt of the previous task as the beginning element prompt of the next task. To verify this, we visualized two attention heads from the T5’s multiple attention heads in Figure 3. In this example, AOPE and E2E-ABSA are used as basic tasks, and ASTE is used as advanced task. Through the analysis of decoder-attention visualization, we have following findings.

Some attention heads learn associations between  $a$  and  $s$ . As shown in Figure 3(a), <extra\_id\_1> nearly never attend to opinion term(good) and <extra\_id\_2> attend to <extra\_id\_1> heavily where the association of aspect and sentiment is established. Such attention head models the relation between  $a$  and  $s$ .

Some other attention heads learn associations between element  $o$  and  $a$ . As shown in Figure 3(b), the attention weight between <extra\_id\_1> and <extra\_id\_0> is high, which means that the information of the opinion is used when the aspect is generated via the <extra\_id\_1>. Such attention head models the attention relationship between  $o$  and  $a$ .

In a word, combining information from multiple attention heads with different functions, our framework can model advanced tasks through basic tasks.

### LEGO split

This section introduces how to make the framework trained on advanced tasks capable of extracting any custom elements by changing the task prompt like an assembled Lego can be divided into parts of different sizes.

We explored the ASTE task as the target advance task and traverse the full permutation of the three



element prompts of a, o, and s. For each permutation of element prompts, we generate a dataset with specific task prompt that assembled by element prompts. Finally we mix and shuffle all the datasets and train the framework with the setting of multitask training.

As shown in Table 8, we can arbitrarily extract any single element and any combination of elements by changing the task prompt. The framework can perfectly control the output content through the task prompt. This result shows that the approach proposed in this paper can make T5 regard the task prompt as a combination of multiple element prompts, rather than a simple string.

## 5 Conclusion

In this paper, we propose a prompt-based generative framework LEGO-ABSA for ABSA tasks that use T5 as the backbone, which can make full use of the information learned from the T5 unsupervised training object through the formulation of task prompts we proposed.

LEGO-ABSA does not regard the prompt and the output text as a simple string but a combination of multiple elements to be extracted. It is mainly used in multi-task training and task transfer scenarios. Extensive experiments on six ABSA tasks verify the effectiveness of our framework and its excellent transferability in task transfer scenarios. There is still space for improvement in our framework, such as completing the combination extraction of multiple elements task through the learning of single element tasks.

## 6 Acknowledge

Supported by the National Key Research and Development Program of China (No. 2020YFC0833301)

## References

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.

Hongjie Cai, Rui Xia, and Jianfei Yu. 2021. [Aspect-category-opinion-sentiment quadruple extraction with implicit aspects and opinions](#). In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing*

(*Volume 1: Long Papers*), pages 340–350, Online. Association for Computational Linguistics.

- Shaowei Chen, Jie Liu, Yu Wang, Wenzheng Zhang, and Ziming Chi. 2020. Synchronous double-channel recurrent network for aspect-opinion pair extraction. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6515–6524.
- Shaowei Chen, Yu Wang, Jie Liu, and Yuelin Wang. 2021. Bidirectional machine reading comprehension for aspect sentiment triplet extraction. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 12666–12674.
- Zehui Dai, Cheng Peng, Huajie Chen, and Yadong Ding. 2020. [A multi-task incremental learning framework with category name embedding for aspect-category sentiment analysis](#).
- Lei Gao, Yulong Wang, Tongcun Liu, Jingyu Wang, Lei Zhang, and Jianxin Liao. 2021. Question-driven span labeling model for aspect-opinion pair extraction. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 12875–12883.
- 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.
- Ehsan Hosseini-Asl, Wenhao Liu, and Caiming Xiong. 2022. A generative language model for few-shot aspect-based sentiment analysis. *arXiv preprint arXiv:2204.05356*.
- Mengting Hu, Shiwan Zhao, Li Zhang, Keke Cai, Zhong Su, Renhong Cheng, and Xiaowei Shen. 2018. [Can: Constrained attention networks for multi-aspect sentiment analysis](#).
- Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Ves Stoyanov, and Luke Zettlemoyer. 2019. [Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension](#).
- Xin Li, Lidong Bing, Piji Li, and Wai Lam. 2019a. A unified model for opinion target extraction and target sentiment prediction. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, pages 6714–6721.
- Xin Li, Lidong Bing, Wenxuan Zhang, and Wai Lam. 2019b. [Exploiting BERT for end-to-end aspect-based sentiment analysis](#). In *Proceedings of the 5th Workshop on Noisy User-generated Text (W-NUT 2019)*, pages 34–41.
- Jian Liu, Zhiyang Teng, Leyang Cui, Hanmeng Liu, and Yue Zhang. 2021. Solving aspect category sentiment analysis as a text generation task. *arXiv preprint arXiv:2110.07310*.

- Yukun Ma, Haiyun Peng, and Erik Cambria. 2018. Targeted aspect-based sentiment analysis via embedding commonsense knowledge into an attentive lstm. *Proceedings of the AAAI Conference on Artificial Intelligence*, 32(1).
- Yue Mao, Yi Shen, Chao Yu, and Longjun Cai. 2021. A joint training dual-mrc framework for aspect based sentiment analysis.
- Bonan Min, Hayley Ross, Elier Sulem, Amir Poursan Ben Veyseh, Thien Huu Nguyen, Oscar Sainz, Eneko Agirre, Ilana Heinz, and Dan Roth. 2021. Recent advances in natural language processing via large pre-trained language models: A survey. *arXiv preprint arXiv:2111.01243*.
- Ambreen Nazir, Yuan Rao, Lianwei Wu, and Ling Sun. 2020. Issues and challenges of aspect-based sentiment analysis: A comprehensive survey. *IEEE Transactions on Affective Computing*, pages 1–1.
- Haiyun Peng, Lu Xu, Lidong Bing, Fei Huang, Wei Lu, and Luo Si. 2020. Knowing what, how and why: A near complete solution for aspect-based sentiment analysis. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 8600–8607.
- Maria Pontiki, Dimitris Galanis, Haris Papageorgiou, Ion Androutsopoulos, Suresh Manandhar, Mohammad AL-Smadi, Mahmoud Al-Ayyoub, Yanyan Zhao, Bing Qin, Orphée De Clercq, Véronique Hoste, Marianna Apidianaki, Xavier Tannier, Natalia Loukachevitch, Evgeniy Kotelnikov, Nuria Bel, Salud María Jiménez-Zafra, and Gülşen Eryiğit. 2016. *SemEval-2016 task 5: Aspect based sentiment analysis*. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pages 19–30, San Diego, California. Association for Computational Linguistics.
- Maria Pontiki, Dimitris Galanis, Haris Papageorgiou, Suresh Manandhar, and Ion Androutsopoulos. 2015. *SemEval-2015 task 12: Aspect based sentiment analysis*. In *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)*, pages 486–495, Denver, Colorado. Association for Computational Linguistics.
- Maria Pontiki, Dimitris Galanis, John Pavlopoulos, Haris Papageorgiou, Ion Androutsopoulos, and Suresh Manandhar. 2014. *SemEval-2014 task 4: Aspect based sentiment analysis*. In *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*, pages 27–35, Dublin, Ireland. Association for Computational Linguistics.
- Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners.
- Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67.
- Kim Schouten and Flavius Frasincar. 2016. Survey on aspect-level sentiment analysis. *IEEE Transactions on Knowledge and Data Engineering*, 28(3):813–830.
- Hai Wan, Yufei Yang, Jianfeng Du, Yanan Liu, Kunxun Qi, and Jeff Z Pan. 2020. Target-aspect-sentiment joint detection for aspect-based sentiment analysis. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 9122–9129.
- Wenya Wang and Sinno Jialin Pan. 2018. Recursive 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. 2017. Coupled multi-layer attentions for co-extraction of aspect and opinion terms. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 31.
- Chao Wu, Qingyu Xiong, Hualing Yi, Yang Yu, Qiwu Zhu, Min Gao, and Jie Chen. 2021. Multiple-element joint detection for aspect-based sentiment analysis. *Knowledge-Based Systems*, 223:107073.
- Meixi Wu, Wenya Wang, and Sinno Jialin Pan. 2020a. Deep Weighted MaxSAT for Aspect-based Opinion Extraction. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5618–5628, Online. Association for Computational Linguistics.
- Zhen Wu, Chengcan Ying, Fei Zhao, Zhifang Fan, Xinyu Dai, and Rui Xia. 2020b. Grid tagging scheme for aspect-oriented fine-grained opinion extraction. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 2576–2585.
- Lu Xu, Yew Ken Chia, and Lidong Bing. 2021. Learning span-level interactions for aspect sentiment triplet extraction.
- Lu Xu, Hao Li, Wei Lu, and Lidong Bing. 2020. Position-aware tagging for aspect sentiment triplet extraction. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2339–2349, Online. Association for Computational Linguistics.
- Hang Yan, Junqi Dai, Tuo Ji, Xipeng Qiu, and Zheng Zhang. 2021. A unified generative framework for aspect-based sentiment analysis. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 2416–2429, Online. Association for Computational Linguistics.

Chen Zhang, Qiuchi Li, Dawei Song, and Benyou Wang. 2020. A multi-task learning framework for opinion triplet extraction. *arXiv preprint arXiv:2010.01512*.

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

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