MvP: Multi-view Prompting Improves Aspect Sentiment Tuple Prediction

Zhibin Gou[*], **Qingyan Guo**[*], **Yujiu Yang**[†]
Tsinghua University
zebgou@gmail.com gqy22@mails.tsinghua.edu.cn
yang.yujiu@sz.tsinghua.edu.cn

Abstract

Generative methods greatly promote aspect-based sentiment analysis via generating a sequence of sentiment elements in a specified format. However, existing studies usually predict sentiment elements in a fixed order, which ignores the effect of the interdependence of the elements in a sentiment tuple and the diversity of language expression on the results. In this work, we propose ***Multi-view Prompting*** (MvP) that aggregates sentiment elements generated in different orders, leveraging the intuition of human-like problem-solving processes from different views. Specifically, MvP introduces element order prompts to guide the language model to generate multiple sentiment tuples, each with a different element order, and then selects the most reasonable tuples by voting. MvP can naturally model multi-view and multi-task as permutations and combinations of elements, respectively, outperforming previous task-specific designed methods on multiple ABSA tasks with a single model. Extensive experiments show that MvP significantly advances the state-of-the-art performance on 10 datasets of 4 benchmark tasks, and performs quite effectively in low-resource settings. Detailed evaluation verified the effectiveness, flexibility, and cross-task transferability of MvP.[1]

1 Introduction

Aspect-based sentiment analysis (ABSA) aims to predict tuples of sentiment elements of interest for a given text. There are four sentiment elements that constitute the main line of ABSA research: aspect term (a), aspect category (c), opinion term (o) and sentiment polarity (s) (Zhang et al., 2022). Given an example sentence, "I love the sushi badly!", the corresponding elements are "sushi", "food quality", "love" and "positive", respectively. Early studies focus on a single sentiment element like aspect term (Liu et al., 2015; Ma et al., 2019), aspect category (Zhou et al., 2015) or sentiment polarity (Wang et al., 2016; Chen et al., 2017). Recent works propose compound ABSA tasks involving multiple associated elements, such as aspect sentiment triplet extraction (ASTE) (Peng et al., 2020), target aspect sentiment detection (TASD) (Wan et al., 2020), aspect sentiment quad prediction (ASQP) (Zhang et al., 2021a) and aspect category opinion sentiment (ACOS) (Cai et al., 2020a). Their target formats are shown in Table 1.

Task	Output
Aspect Category Opinion Sentiment (ACOS)	(a, c, o, s)
Aspect Sentiment Quad Prediction (ASQP)	(a, c, o, s)
Aspect Sentiment Triplet Extraction (ASTE)	(a, o, s)
Target Aspect Sentiment Detection (TASD)	(a, c, s)

Table 1: Aspect sentiment tuple prediction tasks with their corresponding outputs. Notably, although both ACOS and ASQP are the most complex quadratic prediction tasks, ACOS focuses on implicit aspects and opinions compared to ASQP. Detailed tasks and dataset statistics are shown in Appendix A.

Recently, generative methods have been used to handle various ABSA tasks uniformly and achieved good performance (Zhang et al., 2022), where the common practice is to generate a sequence of sentiment elements in a specified format to leverage label semantics. To be specific, they use class index (Yan et al., 2021), sentiment element sequence (Zhang et al., 2021d), natural language (Liu et al., 2021a; Zhang et al., 2021b), structured extraction schema (Lu et al., 2022b) or opinion tree (Bao et al., 2022) as the target of the generation models.

However, previous works usually generate the sequence of sentiment elements in a left-to-right fixed order, which ignores the influence of the interdependence of the elements in a sentiment tuple

[*]Equal contribution.
[†]Corresponding author.
[1]Code and data released at https://github.com/ZubinGou/multi-view-prompting

Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics
Volume 1: Long Papers, pages 4380–4397
July 9-14, 2023

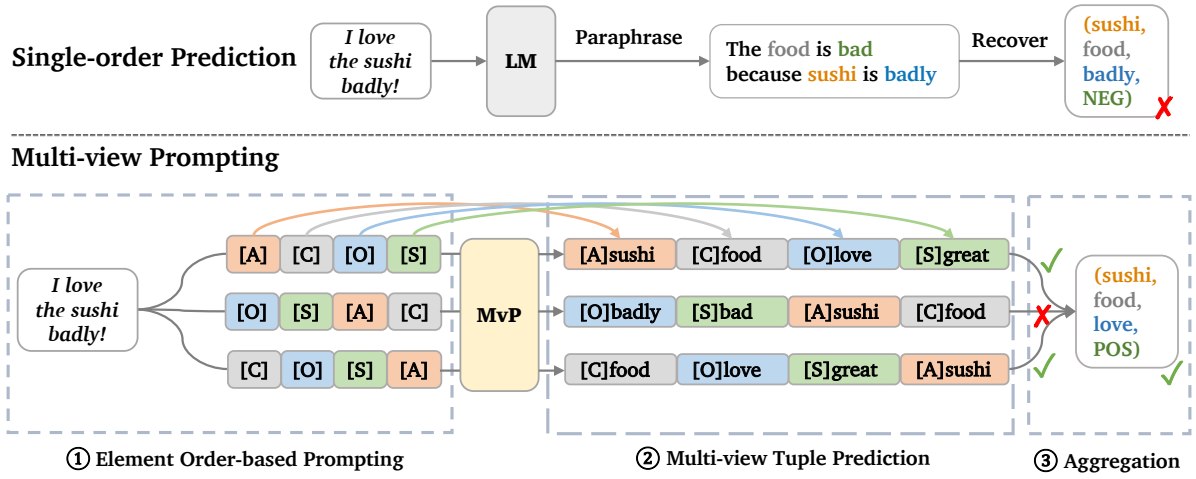

Figure 1: Compared with predicting in a single order, MvP proposes element-order prompt learning to control the prediction order of sentiment element. MvP contains three steps: ① permutes multiple elements to form order prompts and constructs an appropriate subset in terms of conditional generation scores; ② generates multiple sequences consisting of tuples from different views based on the prompt subset. The element order of each tuple accords with the prompt in the input; ③ aggregates the multiple predictions and obtains the final output.

and the diversity of language expression on the targets. For example, the "$c \Rightarrow s \Rightarrow a \Rightarrow o$" order in PARAPHRASE (Zhang et al., 2021b) (Figure 1). This single-order generation has the following potential drawbacks: (1) Incompleteness, tuple prediction is not naturally a text generation task, the relationship among elements is not ordered but interdependent; (2) Instability, as shown in a study by Hu et al. (2022), the performance of different target template orders differs significantly; (3) Error accumulation, the previous prediction errors will be accumulated and affect later predictions.

To address the above challenges, we propose ***M**ulti-view **P**rompting* (MvP) that aggregates sentiment elements predicted in different orders, leveraging the intuition of solving problems from different views in human reasoning and decision (Stanovich and West, 2000). Inspired by prompt chaining (Liu et al., 2021b; Wei et al., 2022b; Wang et al., 2022b,a), MvP introduces element order-based prompt learning to control the prediction order of sentiment elements, enabling diverse target expressions. Compared to single-order generation, MvP mitigates the incompleteness and instability of a fixed order by receiving information from multiple views, while alleviating the potential error accumulation of generative methods via permutation of elements (Figure 1). Besides, MvP is naturally suited for training a single model to solve multiple ABSA tasks as combinations of elements, adaptively enabling knowledge transfer from related tuple prediction tasks.

We conduct extensive experiments on main aspect sentiment tuple prediction tasks, including ASQP, ACOS, ASTE and TASD. Empirical results show the superiority of MvP in supervised, low-resource, and cross-task transfer settings. In supervised settings, the single-task and multi-task MvP outperform the state-of-the-art by 1.34% and 1.69% absolute F1 scores on all tasks, respectively. At low resource settings, MvP has sizable improvement over strong baselines, and cross-task transfer brings a more remarkable improvement.

Our major contributions are as follows:

1) We introduce MvP, an element order-based prompt learning method that improves sentiment tuple prediction by aggregating multi-view results.

2) MvP naturally allows us to train a single model simultaneously on all tasks. To the best of our knowledge, the multi-tasking MvP is the first single model that substantially outperforms task-specific models on various ABSA tasks.

3) Experiments show that MvP significantly advances the state-of-the-art on 10 datasets of 4 tasks and is quite effective in low-resource settings.

2 Methodology

To better understand the operation process of the proposed MvP, we can carefully observe the pipeline shown in Figure 1. Unlike the fixed order element prediction adopted by previous methods like PARAPHRASE (Zhang et al., 2021b), we take

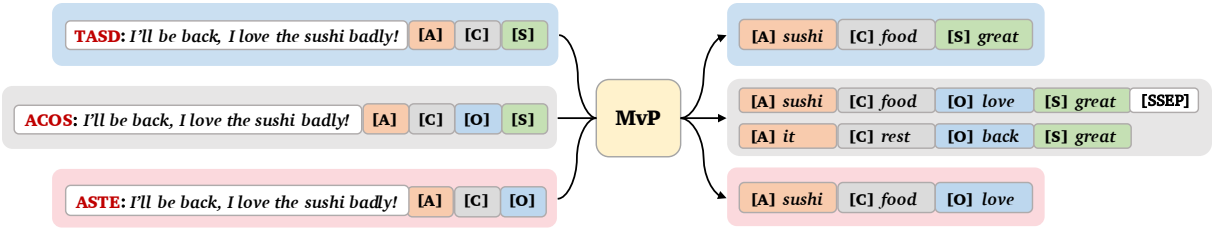

Figure 2: Multi-task learning. MvP uniformly tackles ABSA tasks as combination of element order prompts.

every possible permutation of sentiment elements (6 for the triplet and 24 for the quadruplet) into account and select the appropriate subsets of them for efficiency and effectiveness reasons. Conditioned on different ordered prompts, a model can generate multiple tuples from different views. Some views give the same correct tuples, while some views are less effective and thus might be wrong, but it's unlikely to result in the same error. In other words, different views tend to show more agreement in the correct sentiment tuples. Following this intuition, the proposed MvP aggregates and takes the tuples that most views agree on as the final result.

2.1 Problem Definition

In this section, we present our approach with the quadruple task by default, which can be applied to triplet tasks with minor modifications. We formally define the task as follows:

Given an input sentence, aspect sentiment tuple prediction aims to predict all sentiment tuples $T = \{(a, c, o, s)\}$, each consisting of aspect term (a), aspect category (c), opinion term (o) and sentiment polarity (s). To leverage label semantics, following previous works (Zhang et al., 2021a), we paraphrase these elements to natural language e_a, e_c, e_o, e_s separately. For example, we map the "POS" label of sentiment polarity s to "great", and map the "NULL" label of opinion term o to "it".

2.2 Element Order-based Prompt Learning

To control the prediction order of sentiment elements, MvP introduces an element order-based prompting mechanism. Specifically, we design the target with ordered target schema and input with element order prompts.

2.2.1 Ordered Target Schema

To indicate different sentiment elements, we follow the DLO method (Hu et al., 2022) and design element markers to represent the structure of the information (Paolini et al., 2021). The element markers for e_a, e_c, e_o, and e_s are [A], [C], [O] and [S], respectively. We add the corresponding marker as a prefix to each element and concatenate them in a given permutation p_i as the target sequence, for example, "[O]e_o[A]e_a[C]e_c[S]e_s".

If there are multiple sentiment tuples for an input sentence, we utilize a special symbol [SSEP] to concatenate their corresponding ordered target schema to get the final target sequence $\boldsymbol{y}_{\boldsymbol{p_i}}$.

2.2.2 Element Order Prompts

We design element order prompts by concatenating these element markers to represent the desired order p_i of sentiment elements (for example, "[O][A][C][S]" indicates prediction in the order of "$o \Rightarrow a \Rightarrow c \Rightarrow s$"). Then, we add the prompt as a suffix to each input sentence to get the final input $\boldsymbol{x}_{\boldsymbol{p_i}}$. Thus we obtain an input-output pair for training:

Input **($\boldsymbol{x}$):** I love the sushi badly! [O][A][C][S]

Output **($\boldsymbol{y}$):** [O] love [A] sushi [C] food [S] great

We find that the design of element order prompts can effectively guide sentiment tuples' generation order. Thus multi-view and multi-task can be flexibly modeled through the permutation and combination of elements.

2.3 Multi-view Training

For training, MvP selects appropriate element orders to construct input-target pairs, and then fine-tunes a Seq2Seq model.

2.3.1 Element Order Selection

Since overheads increase linearly with the number of views and the performance of different views varies, we need to select appropriate element orders. Following the study of prompt ordering (Lu et al., 2022a; Hu et al., 2022), we choose the potentially better-performing orders based on the average entropy of the candidate permutations on the training set. The steps are as follows: (i) we use every possible permutation p_i of sentiment elements as candidates; (ii) given an input sentence $\boldsymbol{x}$ and its target tuples, we construct the ordered target schema

$\boldsymbol{y_{p_i}}$ of permutation p_i as described in §2.2.1, replace the element markers in it with spaces to avoid noises, and query a pre-trained language model to get conditional generation scores $p(\boldsymbol{y_{p_i}}|\boldsymbol{x})$; and (iii) calculate the average score of permutation p_i over the training set D:

$$S_{p_i} = \frac{\sum_D p(\boldsymbol{y_{p_i}}|\boldsymbol{x})}{|D|} \quad (1)$$

Thus we can rank each permutation p_i with S_{p_i} and top m permutations are used for training.

2.3.2 Training

With the selected m permutations, we construct m different ordered prompts and targets for each sentence. Given the input-target pair $(\boldsymbol{x}, \boldsymbol{y})$, we can fine-tune a pre-trained sequence-to-sequence language model (LM) such as BART (Lewis et al., 2020) or T5 (Raffel et al., 2020), minimizing the following negative log-likelihood loss:

$$\begin{aligned} \mathcal{L}_{NLL} &= -\mathbb{E} \log p(\boldsymbol{y}|\boldsymbol{x}) \\ &= -\mathbb{E} \sum_{t=1}^{T} \log p(\boldsymbol{y_t}|\boldsymbol{x}, \boldsymbol{y_{<t}}) \end{aligned} \quad (2)$$

where T is the length of the target sequence $\boldsymbol{y}$ and $\boldsymbol{y_{<t}}$ denotes previously generated tokens.

2.4 Multi-view Inference

For inference, MVP prompts the trained model to normatively generate multiple sentiment tuples in previously selected orders, and finally aggregate to obtain the most reasonable tuples.

2.4.1 Schema Constrained Generation

Given an input sentence, we construct multiple prompts in the same order as the training, which guides the model to generate targets from different views. However, the generated results may not conform to the target schema format, especially when the training set is small (Zhang et al., 2021a; Yan et al., 2021). Therefore, we designed a schema-based constrained decoding (Cao et al., 2021) that injects target schema knowledge into the decoding process. It ensures that the generated elements are in the corresponding vocabulary set. See Appendix B for implementation details.

2.4.2 Multi-view Results Aggregation

Since each view may predict more than one tuple, we first aggregate the results of all views and then use the tuples that appear in most views as the final prediction. Specifically, for an input sentence $\boldsymbol{x}$, suppose we prompt a trained model to generate from m selected permutations, and the set of predicted tuples for permutation p_i is T'_{p_i}, which may contain one or more sentiment tuples, and then we can obtain the final aggregated result T'_{MVP} by the following equation:

$$T'_{\text{MVP}} = \{t | t \in \bigcup_{i=1}^{m} T'_{p_i} \text{ and } (\sum_{i=1}^{m} \mathbb{1}_{T'_{p_i}}(t) \geq \frac{m}{2})\}$$

3 Experiments

3.1 Tasks and Dataset

We validate our methods on 10 datasets over 4 tasks, including quadruplet tasks, ASQP and ACOS, and triplet tasks, ASTE and TASD. For a fair comparison, we apply the same data splits as previous works. The targets of each task are shown in Table 1 and the detailed statistics are in Appendix A.

For the ASQP task, we adopt two datasets in the restaurant domain based on SemEval tasks (Pontiki et al., 2015, 2016), Rest15 and Rest16 aligned and completed by Zhang et al. (2021a) subsequently. For the ACOS task, we apply Restaurant-ACOS and Laptop-ACOS constructed by Cai et al. (2021). Compared with ASQP, datasets of ACOS focus on implicit aspects and opinions, which helps to measure our methods comprehensively. For the triple tasks, we adopt datasets provided by Xu et al. (2020) and Wan et al. (2020) for ASTE (Peng et al., 2020) and TASD, respectively.

3.2 Implement Details

We employ T5-BASE model (Raffel et al., 2020) from Huggingface Transformers library[2](Wolf et al., 2020) as the pre-trained model. T5 adopts a classical encoder-decoder architecture similar to Transformer (Vaswani et al., 2017). We use greedy search for decoding by default. We use the same hyperparameters across all tasks and datasets, and detailed settings can be found in Appendix C.

The number of views m is set to 5 by default across the majority of the experiments, including multi-task, low-resource, cross-task transfer, and ablations. Only the single-task model in the main experiment uses 15 views for the quadruplet tasks and 5 views for the triplet tasks. For simplicity, the number of views in inference is the same as that in

[2] https://github.com/huggingface/transformers

Methods	ASQP		ACOS		TASD		ASTE				AVG
	R15	R16	Lap	Rest	R15	R16	L14	R14	R15	R16	
TAS-BERT (Wan et al., 2020)	34.78	43.71	27.31	33.53	57.51	65.89	-	-	-	-	-
Jet-BERT (Xu et al., 2020)	-	-	-	-	-	-	51.04	62.40	57.53	63.83	-
Extract-Classify (Cai et al., 2021)	36.42	43.77	35.80	44.61	-	-	-	-	-	-	-
GAS (Zhang et al., 2021c)	45.98	56.04	-	-	60.63	68.31	58.19	70.52	60.23	69.05	-
Paraphrase (Zhang et al., 2021b)	46.93	57.93	43.51	61.16	63.06	71.97	61.13	72.03	62.56	71.70	61.20
UIE (Lu et al., 2022b)	-	-	-	-	-	-	62.94	72.55	64.41	72.86	-
Seq2Path (Mao et al., 2022)	-	-	42.97	58.41	63.89	69.23	64.82	75.52	65.88	72.87	-
DLO (Hu et al., 2022)	48.18	59.79	43.64	59.99	62.95	71.79	61.46	72.39	64.26	73.03	61.75
UnifiedABSA† (Wang et al., 2022c)	-	-	42.58	60.60	-	-	-	-	-	-	-
LEGO-ABSA† (Gao et al., 2022)	46.10	57.60	-	-	62.30	71.80	62.20	73.70	64.40	69.90	-
MvP	51.04	**60.39**	**43.92**	**61.54**	64.53	**72.76**	63.33	74.05	65.89	**73.48**	63.09
MvP (multi-task)†	**52.21**	58.94	43.84	60.36	**64.74**	70.18	**65.30**	**76.30**	**69.44**	73.10	**63.44**

Table 2: Main results on 10 datasets of ASQP, ACOS, TASD and ASTE tasks. F1 scores are reported; the best results are in bold, while the second best are underlined. † indicates multi-tasking models.

training. The case of using a different number of views is left for further exploration.

In the multi-task settings, to introduce domain information, we simply add the task name and dataset name followed by colon separators (e.g. "ASQP: Rest15: ") as the prefixes to each input sentence, and train a single model on all datasets across all tasks (ASQP, ACOS, TASD, ASTE). We select the appropriate element orders for each dataset separately. We find overlap between the input sentences across different splits of different datasets. Therefore, to avoid data leakage, we collect the training sets from all datasets, and discard samples that overlap with the test set of any task. Then we split the data by 9:1 to obtain the final training and validation sets for our multi-tasking method.

3.3 Evaluation Metrics

For all ABSA tasks, a predicted sentiment tuple is considered as correct if and only if all its elements are exactly the same as the gold tuple. We use F1 scores as the main evaluation metrics (Zhang et al., 2021a; Mao et al., 2022). All reported F1 scores are averaged over 5 runs with different random seeds. For multi-task settings, we use a different split of the training and development sets in each run.

3.4 Compared Methods

We compare our methods with the following three types of previous state-of-the-art methods:

Discriminative methods. **TAS-BERT**, based on extraction, (Wan et al., 2020) jointly detects the sentiment tuples. **Extract-Classify** (Cai et al., 2021) decomposes the ACOS task into two steps. For ASTE, **Jet-BERT** (Xu et al., 2020) addresses the task in an end-to-end framework by a tagging scheme.

Generative methods. **GAS** (Zhang et al., 2021c) is the first to model ABSA tasks as a generation process. **Paraphrase** (Zhang et al., 2021a) designs semantic templates filled with fixed-order elements of tuples as generation targets. **Seq2Path** (Mao et al., 2022) generates tuples as paths of a tree and then selects valid ones. **DLO / ILO** (Hu et al., 2022) augments ASQP dataset given the order-free property of the quadruplet based on templates. We also consider **UIE** (Lu et al., 2022b), a unified text-to-structure framework to model various IE tasks which is pre-trained on large-scale data.

Multi-tasking methods. A recent trend is tackling multiple ABSA tasks uniformly using a single multi-tasking model. **LEGO-ABSA** (Gao et al., 2022) designs task prompts similar to T5 and **UnifiedABSA** (Wang et al., 2022c) adopts instruction tuning (Mishra et al., 2022; Wei et al., 2022a).

As a fair comparison, all results of these supervised methods are obtained from the base pre-trained model, either BERT or T5.

Large language model (LLM). To compare our method with advanced large language models, we additionally include evaluation results of **ChatGPT**[3] (`gpt-3.5-turbo`) on the four ABSA tasks with zero- and few-shot prompts. The results can be found in Appendix D.

[3] https://chat.openai.com/

	Methods	Transfer Source	1%	2%	5%	10%	20%	AVG
ASQP (R15)	Paraphrase (Zhang et al., 2021b)	-	5.90	15.73	24.16	31.33	37.47	22.92
	DLO (Hu et al., 2022)	-	10.03	15.94	29.13	35.89	40.34	26.27
	MvP	-	**13.46**	**22.58**	**32.44**	**38.48**	**41.82**	**29.76**
	DLO (transfer)	ASTE (R15)	26.28	28.72	35.94	39.48	42.92	34.67
	MvP (transfer)	ASTE (R15)	**28.69**	**33.93**	**40.08**	**43.10**	**45.09**	**38.18**
ACOS (Rest)	Paraphrase (Zhang et al., 2021b)	-	14.85	24.81	38.33	45.32	49.64	34.59
	DLO (Hu et al., 2022)	-	19.84	29.84	38.47	43.45	46.47	35.61
	MvP	-	**23.84**	**32.57**	**42.89**	**47.77**	**53.54**	**40.12**
	DLO (transfer)	ASTE (R16)	31.06	40.55	43.23	45.74	47.98	41.71
	MvP (transfer)	ASTE (R16)	**39.24**	**42.72**	**49.78**	**52.53**	**55.28**	**47.91**
TASD (R16)	Paraphrase (Zhang et al., 2021b)	-	26.29	36.70	49.48	55.66	61.79	45.98
	DLO (Hu et al., 2022)	-	29.66	41.17	50.44	58.27	62.43	48.39
	MvP	-	**34.00**	**41.76**	**52.58**	**58.93**	**64.53**	**50.36**
	DLO (transfer)	ASQP (R16)	66.25	66.21	64.54	67.99	68.50	66.70
	MvP (transfer)	ASQP (R16)	**68.49**	**68.06**	**68.47**	**68.98**	**69.89**	**68.78**
ASTE (L14)	Paraphrase (Zhang et al., 2021b)	-	16.29	29.20	38.61	45.20	52.88	36.44
	DLO (Hu et al., 2022)	-	17.07	26.07	38.92	48.85	53.82	36.95
	MvP	-	**28.17**	**34.38**	**42.89**	**52.33**	**54.60**	**42.47**
	DLO (transfer)‡	ASQP (R16)	44.76	48.86	51.22	**56.43**	56.71	51.60
	MvP (transfer)‡	ASQP (R16)	**48.43**	**50.33**	**54.27**	56.34	**59.05**	**53.68**

Table 3: Low-resource and cross-task transfer results. We cover 4 tasks, 4 datasets and 2 domains. For a fair comparison, here we choose DLO-top5 which augments the original training set by 5 times. In cross-task transfer settings, for quadruplet tasks ASQP and ACOS, we first train the model on ASTE (R15) and ASTE (R16), respectively, while for triplet tasks TASD and ASTE, the transfer source is ASQP (R16). Then we vary the percentage of transfer target training set and report the results. ‡ It is notable that transferring setting on ASTE (L14) is cross-domain, from Restaurant to Laptop.

4 Results and Discussions

4.1 Single-task and Multi-task Results

Our methods outperform previous best baselines significantly in supervised settings among 4 tasks, 10 datasets, becoming the new state-of-the-art in all of them. As shown in Table 2, we observe that:

1) *By aggregating results from multiple views,* MvP *surpasses the most of previous single-order methods.* In comparison with Paraphrase, the single-view method which applies templates with elements in a fixed order, MvP achieves a sizable improvement of 1.89% on average, verifying the effectiveness of multiple informative views.

2) *Element order-based prompt learning effectively guides the generation of tuples by unifying training and inference.* Compared with DLO augmenting data on the target side, MvP obtains an improvement of 1.34% by generating tuples controllably with designed element order-based prompts.

3) MvP *can be applied without abundant pre-training simply and achieves better performance.* It is notable that MvP trained on T5-base exceeds UIE using T5-v1.1-base and subsequently trained on a large corpus with 65M instances on all datasets of ASTE (+1.00 on average).

Multi-task Learning. *By permutation and combination,* MvP *(multi-task) obtains generalized ability among diverse tasks.* Compared with LEGO-ABSA, a multi-task unified baseline, MvP (multi-task) obtains a +2.82% absolute improvement in F1 score on average.

MvP, a single model completing tuple prediction by information from multiple views, can be naturally employed on multiple ABSA tasks, achieving competitive or better performances than previous methods that require task-specific fine-tuning, data augmentation (e.g., Seq2Path) or complex pre-

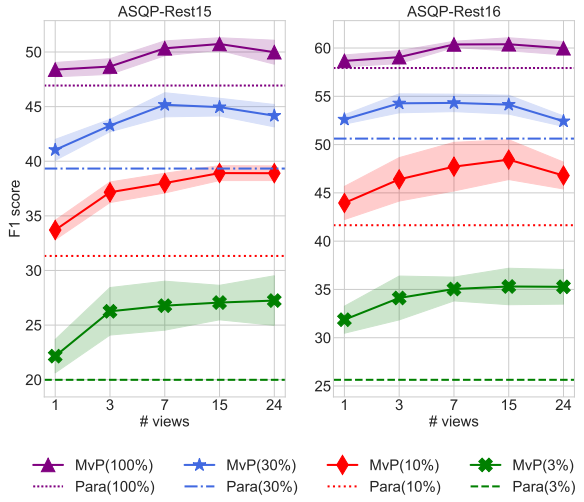

Figure 3: Effect of the number of views. "# views" refers to the number of views used for training and inference, and "Para" stands for Paraphrase. Values in parentheses represent the ratio of the training data used.

training (e.g., UIE), either in single-task or multi-tasking settings, showing strong stability.

4.2 Low-resource Results

To further explore the behavior of our methods in low-resource settings, we train Paraphrase, DLO, MvP using 1%, 2%, 5%, 10%, and 20% of 4 different training sets in 2 domains over 4 tasks. The F1 scores of test sets are reported in Table 3. We find that MvP, with efficient prompts from different views, achieves better results than previous works with only a small number of samples. In particular, MvP outperforms Paraphrase and DLO substantially in all settings with a performance boost of 5.70% and 3.87% F1 on average.

Cross-task transfer. *Based on* MvP*, transfer brings further significant improvements, from triplets to quadruplets and vice versa.* MvP (transfer) performs relatively well in extremely low-resource situations, thus exceeding strong baselines under cross-task transfer situations, both in-domain and cross-domain. Compared with DLO (transfer), MvP (transfer) achieves considerably better results under various transfer settings, showing a strong transferability (from 50.36% to 68.78% for TASD). Rather than capture task-specific features, MvP effectively shares ABSA abilities. MvP (transfer) trained with simple tasks (TASD and ASTE here) with adequate data can be easily transferred to tough tasks (ASQP and ACOS here) when the dataset sizes are small, and vice versa.

Methods	ASTE (L14)			ASQP (R15)		
	1%	10%	100%	1%	10%	100%
MvP w/o cd	21.37	49.98	63.27	12.09	37.87	50.92
MvP (rand)	27.32	51.02	62.50	**13.56**	37.18	49.84
MvP (rank)	25.98	49.98	62.48	13.38	37.45	49.98
MvP	**28.37**	**52.33**	**63.33**	13.46	**38.48**	**51.04**

Table 4: Ablation of constrained decoding and effect of aggregation strategy on ASTE (L14) and ASQP (R15). "w/o cd" discards the constrained decoding during inference. MvP (rank) and MvP(rand) are both single-view strategies. The former selects the top-ranked sequence based on the prediction scores (perplexity) of generated sequences from multiple views during inference while the latter randomly samples one.

4.3 Effect Analysis

Effect of the number of views. MvP raises a question that how many views should be selected for training and inference, which we further explore by varying the number of views and the size of the training set (Figure 3). As the number of views increases, curves show an ascending trend first. Interestingly, when the resource is adequate, F1 decreases slightly after a certain number (between 7 and 15). We believe views with lower ranks may be less effective. Thus it is crucial to balance the size of the data and the number of views. It is dramatic that MvP performs decently even with a single view, probably due to the appropriate order selection and the constrained decoding. In extremely low-resource scenarios, by setting a larger value, MvP can expand single-view information and provide more potential choices. The maximum number of views for quadruplets is much higher than that for triples, making MvP more appropriate for quadruple tasks. We provide further comparisons with single-view prompting (i.e., selecting the best single view for training and inference) on all tasks in the Appendix E.

Effect of aggregation strategy. To explore the effect of different aggregation strategies, we conduct ablation studies mainly on ASTE and ASQP tasks, as shown in Table 4. We can see that replacing majority voting with random selection or ranking results in a reduction of F1 in most cases, indicating that majority voting is a more stable strategy for handling diverse views.

Effect of constrained decoding. The designed constrained decoding guides the generation of different views by limiting the predicted term to a specific list. The impact of this algorithm increases

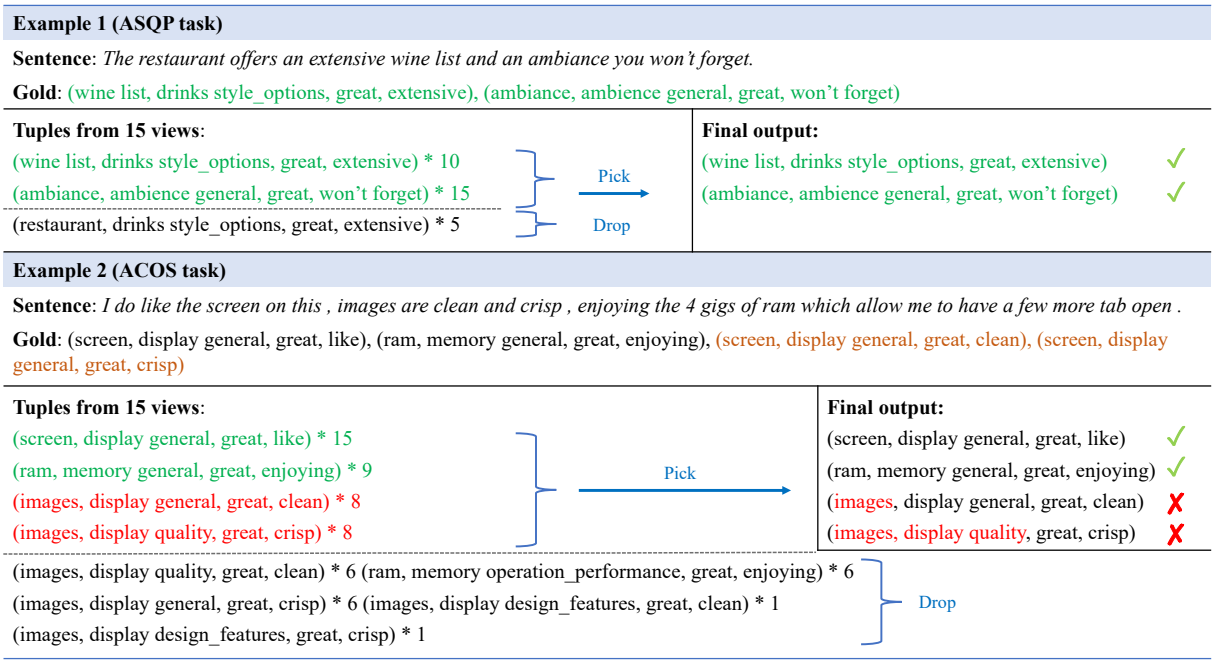

Figure 4: Two examples including the input sentence, quadruplets or triples predicted, and the final outputs of MvP after filtering by voting. *Pick* means that the tuple has appeared in more than half of the predictions in multiple views, while *drop* means that it has appeared less than half of the times and is discarded. Words in green are positive ones while those in red are wrongly picked. Tuples in orange are the ones that MvP ignores.

as the size of the data decreases, and in extremely low-resource scenarios, MvP combined with this algorithm performs considerably well (Table 4).

4.4 Case Study & Error analysis

Figure 4 shows two examples in Rest16 ASQP and Laptop-ACOS, respectively. It can be observed from Example 1 that MvP handles cases with multiple sentiment tuples in a sentence well after filtering unreasonable tuples predicted, i.e. (*restaurant, drinks style_options, great, extensive*), appearing in five generated results in the case. MvP only outputs tuples considered important in most views and thus repairs the error in the single view by receiving and aggregating information from multiple views. In Example 2, the challenging Laptop dataset includes 121 categories, and we can see that while multi-view prompting provides more possibilities and choices, it still confuses similar aspect categories, i.e., *display general* and *display quality*.

5 Related Works

Aspect-base Sentiment Analysis. ABSA has received wide attention in recent years. Early studies focused on extracting or predicting a single sentiment element like aspect term extraction (Qiu et al., 2011; Liu et al., 2015; Ma et al., 2019), aspect category detection (Zhou et al., 2015; Bu et al., 2021) or sentiment polarity classification for a given aspect (Wang et al., 2016; Chen et al., 2017; Lei et al., 2018, 2019). Some works further consider the joint prediction of two associated elements (Cai et al., 2020b), including aspect-opinion pair extraction (Wang et al., 2017; Chen et al., 2020), aspect term-polarity co-extraction (Huang and Carley, 2018; Luo et al., 2019; Chen and Qian, 2020). And recent works propose more challenging ABSA tasks to predict sentiment triplets or quadruplets (Chen et al., 2022), the most influential of which are ASTE (Peng et al., 2020; Zhai et al., 2022), TASD (Wan et al., 2020), ASQP (Zhang et al., 2021a) and ACOS with an emphasis on the implicit aspects or opinions (Cai et al., 2020a).

Generative ABSA. Instead of separate or pipeline methods (Phan and Ogunbona, 2020), most recent works attempt to tackle various ABSA problems using a unified framework (Sun et al., 2022). Generative methods achieve good performance in ABSA by mitigating the potential error propagation in pipeline methods and fully exploiting the rich label semantic information (Paolini et al., 2021; Zhang et al., 2022; Yu et al., 2023). They use sentiment element sequence (Zhang et al., 2021d), natural language (Liu et al., 2021a; Zhang

et al., 2021b) and structured extraction schema (Lu et al., 2022b) etc. as the generative targets. Recently proposed LEGO-ABSA (Gao et al., 2022) and UnifiedABSA (Wang et al., 2022c) focus on multi-tasking with task prompts or instruction design. Hu et al. (2022) firstly investigate element ordering and propose methods to augment target-side data with selected orders for the ASQP task. Despite the promising results, the augmentation may confuse the model with multiple targets for the same input (i.e., one-to-many), thus leading to discrepancies between inference and training. We fill the gap and eliminate such confusion by aligning training and inference with multi-view prompt learning.

6 Conclusion

In this work, we introduce an element order-based prompt learning method - MvP, which improves aspect-level opinion information prediction by simple yet effective multi-view results aggregation. Leveraging the intuition of solving problems from different views, MvP advances the research of generative modeling for tuple structure prediction. By combining and permuting the sentiment elements, our multi-tasking model substantially outperforms task-specific models on a variety of ABSA tasks. Detailed experiments show that our method significantly advances the state-of-the-art on benchmark datasets, in both supervised and low-resource settings. We hope our research will shed light on generative tuple prediction.

Limitations

Despite the state-of-the-art performances, our proposed methods still have some limitations for future directions. **Firstly**, multi-view prompting creates overheads of training and inference proportional to the number of views. For efficiency in practice, according to Figure 3, MvP with a relatively small number of views behaves decently (e.g., 5 or 7). **Secondly**, we apply a simple yet effective aggregation strategy to combine the results of multiple views. More advanced strategies can be explored. **Lastly**, experiments only verified the consistent improvement on ABSA tasks, while intuitively, the idea of MvP that leverages multiple views can be expanded to any structure prediction tasks, such as information extraction, emotion-cause pair extraction, and stance detection.

Ethics Statement

We conduct all the experiments on existing datasets widely used in previous public scientific papers. We keep fair and honest in our analysis of experimental results, and our work does not harm anyone. We open-sourced our code for further explorations.

As for the broader impact, this work may foster further research in sentiment analysis using generative methods, contributing to the simplification and automation of user opinion mining in reality. Nevertheless, this work fine-tunes large pre-trained language models to generate sentiment tuples. Due to the large pre-training corpus based on the Internet, the predicted sentiment polarity is subject to unexpected bias with respect to gender, race, and intersectional identities (Tan and Celis, 2019), which needs to be considered more broadly in the field of natural language processing.

Acknowledgements

We express our gratitude to Jiayi Li and Chengze Yu for their detailed feedback on a draft of the paper. We also thank Junjie Wang and Taiqiang Wu for their helpful discussion on our presentation. This work was partly supported by the National Key Research and Development Program of China (No. 2020YFB1708200), the "Graph Neural Network Project" of Ping An Technology (Shenzhen) Co., Ltd. and AMiner.Shenzhen SciBrain fund.

References

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

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

Hongjie Cai, Yaofeng Tu, Xiangsheng Zhou, Jianfei Yu, and Rui Xia. 2020a. Aspect-category based sentiment analysis with hierarchical graph convolutional network. In *Proceedings of the 28th International Conference on Computational Linguistics, COLING 2020, Barcelona, Spain (Online), December 8-13, 2020*, pages 833–843. International Committee on Computational Linguistics.

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

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

Nicola De Cao, Gautier Izacard, Sebastian Riedel, and Fabio Petroni. 2021. Autoregressive entity retrieval. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net.

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

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.

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, Online. 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.

Tianhao Gao, Jun Fang, Hanyu Liu, Zhiyuan Liu, Chao Liu, Pengzhang Liu, Yongjun Bao, and Weipeng Yan. 2022. LEGO-ABSA: A prompt-based task assemblable unified generative framework for multi-task aspect-based sentiment analysis. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 7002–7012, Gyeongju, Republic of Korea. International Committee on Computational Linguistics.

Mengting Hu, Yike Wu, Hang Gao, Yinhao Bai, and Shiwan Zhao. 2022. Improving aspect sentiment quad prediction via template-order data augmentation. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 7889–7900, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Binxuan Huang and Kathleen Carley. 2018. Parameterized convolutional neural networks for aspect level sentiment classification. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1091–1096, Brussels, Belgium. Association for Computational Linguistics.

Zeyang Lei, Yujiu Yang, Min Yang, and Yi Liu. 2018. A multi-sentiment-resource enhanced attention network for sentiment classification. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 758–763, Melbourne, Australia. Association for Computational Linguistics.

Zeyang Lei, Yujiu Yang, Min Yang, Wei Zhao, Jun Guo, and Yi Liu. 2019. A human-like semantic cognition network for aspect-level sentiment classification. 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 6650–6657. AAAI Press.

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

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

Pengfei Liu, Shafiq Joty, and Helen Meng. 2015. Fine-grained opinion mining with recurrent neural networks and word embeddings. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1433–1443, Lisbon, Portugal. Association for Computational Linguistics.

Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. 2021b. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *CoRR*, abs/2107.13586.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net.

Yao Lu, Max Bartolo, Alastair Moore, Sebastian Riedel, and Pontus Stenetorp. 2022a. Fantastically ordered prompts and where to find them: Overcoming few-shot prompt order sensitivity. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8086–8098, Dublin, Ireland. Association for Computational Linguistics.

Yaojie Lu, Qing Liu, Dai Dai, Xinyan Xiao, Hongyu Lin, Xianpei Han, Le Sun, and Hua Wu. 2022b. Unified structure generation for universal information extraction. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5755–5772, Dublin, Ireland. Association for Computational Linguistics.

Huaishao Luo, Tianrui Li, Bing Liu, and Junbo Zhang. 2019. DOER: Dual cross-shared RNN for aspect term-polarity co-extraction. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 591–601, Florence, Italy. 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.

Yue Mao, Yi Shen, Jingchao Yang, Xiaoying Zhu, and Longjun Cai. 2022. Seq2Path: Generating sentiment tuples as paths of a tree. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 2215–2225, Dublin, Ireland. Association for Computational Linguistics.

Swaroop Mishra, Daniel Khashabi, Chitta Baral, and Hannaneh Hajishirzi. 2022. Cross-task generalization via natural language crowdsourcing instructions. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3470–3487, Dublin, Ireland. Association for Computational Linguistics.

Giovanni Paolini, Ben Athiwaratkun, Jason Krone, Jie Ma, Alessandro Achille, Rishita Anubhai, Cícero Nogueira dos Santos, Bing Xiang, and Stefano Soatto. 2021. Structured prediction as translation between augmented natural languages. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net.

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.

Minh Hieu Phan and Philip O. Ogunbona. 2020. Modelling context and syntactical features for aspect-based sentiment analysis. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 3211–3220, Online. Association for Computational Linguistics.

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.

Guang Qiu, Bing Liu, Jiajun Bu, and Chun Chen. 2011. Opinion word expansion and target extraction through double propagation. *Computational Linguistics*, 37(1):9–27.

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

Keith E Stanovich and Richard F West. 2000. Individual differences in reasoning: Implications for the rationality debate? *Behavioral and brain sciences*, 23(5):645–665.

Tianxiang Sun, Xiangyang Liu, Xipeng Qiu, and Xuan-Jing Huang. 2022. Paradigm shift in natural language processing. *Int. J. Autom. Comput.*, 19(3):169–183.

Yi Chern Tan and L Elisa Celis. 2019. Assessing social and intersectional biases in contextualized word representations. *Advances in Neural Information Processing Systems*, 32.

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.

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 *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 9122–9129. AAAI Press.

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.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc V. Le, Ed H. Chi, and Denny Zhou. 2022a. Rationale-augmented ensembles in language models. *CoRR*, abs/2207.00747.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc V. Le, Ed H. Chi, and Denny Zhou. 2022b. Self-consistency improves chain of thought reasoning in language models. *CoRR*, abs/2203.11171.

Yequan Wang, Minlie Huang, Xiaoyan Zhu, and Li Zhao. 2016. Attention-based LSTM for aspect-level sentiment classification. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 606–615, Austin, Texas. Association for Computational Linguistics.

Zengzhi Wang, Rui Xia, and Jianfei Yu. 2022c. Unified-absa: A unified ABSA framework based on multi-task instruction tuning. *CoRR*, abs/2211.10986.

Jason Wei, Maarten Bosma, Vincent Y. Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M. Dai, and Quoc V. Le. 2022a. Finetuned language models are zero-shot learners. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Ed H. Chi, Quoc Le, and Denny Zhou. 2022b. Chain of thought prompting elicits reasoning in large language models. *Conference on Neural Information Processing Systems (NeurIPS)*.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

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

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

Chengze Yu, Taiqiang Wu, Jiayi Li, Xingyu Bai, and Yujiu Yang. 2023. Syngen: A syntactic plug-and-play module for generative aspect-based sentiment analysis. In *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1–5. IEEE.

Zepeng Zhai, Hao Chen, Fangxiang Feng, Ruifan Li, and Xiaojie Wang. 2022. COM-MRC: A COntext-masked machine reading comprehension framework for aspect sentiment triplet extraction. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 3230–3241, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Wenxuan Zhang, Yang Deng, Xin Li, Lidong Bing, and Wai Lam. 2021a. Aspect-based sentiment analysis in question answering forums. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 4582–4591, Punta Cana, Dominican Republic. Association for Computational Linguistics.

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

Wenxuan Zhang, Xin Li, Yang Deng, Lidong Bing, and Wai Lam. 2022. A survey on aspect-based sentiment

analysis: Tasks, methods, and challenges. *CoRR*, abs/2203.01054.

Yue Zhang, Quan Guo, and Parisa Kordjamshidi. 2021d. Towards navigation by reasoning over spatial configurations. In *Proceedings of Second International Combined Workshop on Spatial Language Understanding and Grounded Communication for Robotics*, pages 42–52, Online. Association for Computational Linguistics.

Xinjie Zhou, Xiaojun Wan, and Jianguo Xiao. 2015. Representation learning for aspect category detection in online reviews. In *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence, January 25-30, 2015, Austin, Texas, USA*, pages 417–424. AAAI Press.

A Data Statistics

Table 9 shows the data statistics of all datasets of the ASQP, ACOS, ASTE and TASD task. For fair comparison, we keep the same train/dev/test division as previous works.

B Constrained Decoding

To make sure the predicted output complies with the mandatory format, we apply the constrained decoding (CD) algorithm in experiments. Rather than search the whole vocabulary for the next token to decode, which may make the model generate invalid sequences that do not match our expectations, CD adjusts the candidate list dynamically in terms of the current state token by token. If the current token is decoded as '[', which means the next token should be selected from a list of terms, i.e., [A], [O], [S] and [C]. Additionally, CD tracks the current term and decodes the next following tokens based on Table 5.

Current Term	Candidate tokens
[A]	Input sentence, [SSEP]
[O]	Input sentence, [SSEP]
[S]	great, bad, neutral, [SSEP]
[C]	All categories, [SSEP]

Table 5: Candidate lists of different terms

C Detailed Experimental Settings

Hyper-parameters for all experiments can be found in Table 6. We employ the AdamW (Loshchilov and Hutter, 2019) as the optimizer. All experiments are carried out with an Nvidia RTX 3090 GPU.

D Comparison with ChatGPT

D.1 Experiments

We refined the prompt design [4] and evaluated ChatGPT (`gpt-3.5-turbo`) on four ABSA tasks. Due to budget constraints, we tested it with 200 random samples for each task.

The experimental results, shown in Table 7, highlight the remarkable performance advantage of cross-task transferred and fully supervised MvP compared to the few-shot prompted ChatGPT (+7.06% and +22.84% absolute F1 scores).

[4]Designed based on https://github.com/RidongHan/Evaluation-of-ChatGPT-on-Information-Extraction.

Hyperparameters	MvP (All supervised)	MvP (Low Resource)			
		1%, 2%, 3%, 5%	10%, 20%	30%	50%
Epoch	20	100	50	30	20
Batch Size	16	8			
Learning Rate	1e-4				

Table 6: Hyper-parameters for all supervised and low-resource settings

Methods	Data	ASQP (R15)	ACOS (Rest)	TASD (R16)	ASTE (L14)
ChatGPT	zero-shot	22.87	27.11	34.08	36.05
ChatGPT	few-shot	34.27	37.71	46.51	38.12
MvP (transfer)	few-shot	28.69	39.24	68.49	48.43
MvP	full-data	**51.04**	**61.54**	**72.76**	**63.33**

Table 7: Comparison with ChatGPT (`gpt-3.5-turbo`). F1 scores are reported. The best results are in bold, while the second best are underlined. The few-shot results of MvP (transfer) are from Table 3.

Methods	ASQP		ACOS		TASD		ASTE				AVG
	R15	R16	Lap	Rest	R15	R16	L14	R14	R15	R16	
SvP (random)	48.32	58.94	43.61	58.16	63.42	71.60	62.36	71.64	62.31	71.59	61.19
SvP (heuristic)	49.02	59.56	43.83	59.38	61.98	71.57	62.09	72.61	65.29	73.27	61.86
SvP (rank)	48.39	58.67	43.86	59.57	62.93	71.26	62.83	72.71	63.57	71.79	61.56
MvP	**51.04**	**60.39**	**43.92**	**61.54**	**64.53**	**72.76**	**63.33**	**74.05**	**65.89**	**73.48**	**63.09**

Table 8: Additional comparison with single-view prompting on 10 datasets of ASQP, ACOS, TASD and ASTE tasks. F1 scores are reported.

D.2 Prompts for ChatGPT

We present zero- and few-shot prompts for ASQP (R15) in Listing 1 and 2. For prompts related to other tasks, please refer to our released code.

E Additional Comparison with Single-view Prompting

Previously, we conducted ablation studies with a single view on several representative tasks and datasets in both full-data and low-resource settings (see §4.3). Figure 3 illustrates that incorporating multiple views leads to significant improvements, especially in low-resource settings. Additionally, Table 4 showcases the performance degradation resulting from the use of two single-view aggregation strategies.

Here, we present additional comparisons with single-view prompting (designated as SvP) on all tasks, i.e., selecting the best single view for training and inference. We experiment using three single-view selection strategies: 1) **random**; 2) **heuristic**: In our pre-experiments, we find that elements ranked ahead of the top selected orders are mostly free-form terms '[A]' and '[O]', which have higher uncertainty than '[C]' and '[S]'. Therefore, we propose using the "[A][O][C][S]" order heuristically; 3) **rank**: choosing a view for each dataset based on the score described in §2.3.1.

As depicted in Table 8, the results show that the SvP methods, whether employing a ranking strategy or a heuristic order chosen through pre-experimentation, exhibit limited improvement over a random order (+0.37 and +0.67). By permuting and aggregating results from multiple views, MvP significantly outperforms SvP across all tasks (+1.90). These consistent improvements further elucidate the efficacy of MvP.

Listing 1: Zero-shot Prompt for ASQP (R15).

According to the following sentiment elements definition:

– The 'aspect term' refers to a specific feature, attribute, or aspect of a product or service that a user may express an opinion about, the aspect term might be 'null' for implicit aspect.
– The 'opinion term' refers to the sentiment or attitude expressed by a user towards a particular aspect or feature of a product or service, the aspect term might be 'null' for implicit opinion.
– The 'aspect category' refers to the category that aspect belongs to, and the available catgories includes: 'location general', ' food prices', 'food quality', 'food general', 'ambience general', 'service general', 'restaurant prices', 'drinks prices', ' restaurant miscellaneous', 'drinks quality', 'drinks style_options', 'restaurant general' and 'food style_options'.
– The 'sentiment polarity' refers to the degree of positivity, negativity or neutrality expressed in the opinion towards a particular aspect or feature of a product or service, and the available polarities inlcudes: 'positive', 'negative' and 'neutral'.

Recognize all sentiment elements with their corresponding aspect terms, aspect categories, opinion terms and sentiment polarity in the following text with the format of [('aspect term', 'opinion term', 'aspect category', 'sentiment polarity'), ...]:

Listing 2: Few-shot Prompt (10 shots) for ASQP (R15).

According to the following sentiment elements definition:

– The 'aspect term' refers to a specific feature, attribute, or aspect of a product or service that a user may express an opinion about, the aspect term might be 'null' for implicit aspect.
– The 'opinion term' refers to the sentiment or attitude expressed by a user towards a particular aspect or feature of a product or service, the aspect term might be 'null' for implicit opinion.
– The 'aspect category' refers to the category that aspect belongs to, and the available catgories includes: 'location general', ' food prices', 'food quality', 'food general', 'ambience general', 'service general', 'restaurant prices', 'drinks prices', ' restaurant miscellaneous', 'drinks quality', 'drinks style_options', 'restaurant general' and 'food style_options'.
– The 'sentiment polarity' refers to the degree of positivity, negativity or neutrality expressed in the opinion towards a particular aspect or feature of a product or service, and the available polarities inlcudes: 'positive', 'negative' and 'neutral'.

Recognize all sentiment elements with their corresponding aspect terms, aspect categories, opinion terms and sentiment polarity in the following text with the format of [('aspect term', 'opinion term', 'aspect category', 'sentiment polarity'), ...]:

Text: never again !
Sentiment Elements: [('null', 'never', 'restaurant general', 'bad')]

Text: the food was mediocre at best but it was the horrible service that made me vow never to go back .
Sentiment Elements: [('food', 'mediocre', 'food quality', 'bad'), ('service', 'horrible', 'service general', 'bad')]

Text: we had the lobster sandwich and it was fantastic .
Sentiment Elements: [('lobster sandwich', 'fantastic', 'food quality', 'great')]

Text: they have it all −− great price , food , and service .
Sentiment Elements: [('null', 'great', 'restaurant prices', 'great'), ('food', 'great', 'food quality', 'great'), ('service', 'great', 'service general', 'great')]

Text: they even scoop it out nice (for those on a diet) not too much not to little .
Sentiment Elements: [('null', 'nice', 'food style_options', 'great')]

Text: also it 's great to have dinner in a very romantic and comfortable place , the service it 's just perfect ... they 're so frendly that we never want to live the place !
Sentiment Elements: [('place', 'romantic', 'ambience general', 'great'), ('place', 'comfortable', 'ambience general', 'great'), ('service', 'perfect', 'service general', 'great')]

Text: my friend from milan and myself were pleasantly surprised when we arrived and everyone spoke italian .
Sentiment Elements: [('null', 'pleasantly surprised', 'restaurant miscellaneous', 'great')]

Text: i had their eggs benedict for brunch , which were the worst in my entire life , i tried removing the hollondaise sauce completely that was how failed it was .
Sentiment Elements: [('eggs benedict', 'worst', 'food quality', 'bad')]

Text: the food is authentic italian – delicious !
Sentiment Elements: [('food', 'authentic italian', 'food quality', 'great'), ('food', 'delicious', 'food quality', 'great')]

Text: a little pricey but it really hits the spot on a sunday morning !
Sentiment Elements: [('null', 'pricey', 'restaurant prices', 'bad'), ('null', 'hits the spot', 'restaurant general', 'great')]

Task	Dataset	#Cat	Train (POS/NEU/NEG)	Dev (POS/NEU/NEG)	Test (POS/NEU/NEG)
ASQP	Rest15	13	834 1,005/34/315	209 252/14/81	537 453/37/305
	Rest16	13	1,264 1,369/62/558	316 341/23/143	544 584/40/177
ACOS	Laptop	121	2,934 2,583/227/1,364	326 279/24/137	816 716/65/380
	Restaurant	13	1,530 1,656/95/733	171 180/12/69	583 668/44/205
ASTE	Laptop14	-	906 817/126/517	219 169/36/141	328 364/63/116
	Rest14	-	1,266 1,692/166/480	310 404/54/119	492 773/66/155
	Rest15	-	605 783/25/205	148 185/11/53	322 317/25/143
	Rest16	-	857 1,015/50/329	210 252/11/76	326 407/29/78
TASD	Rest15	13	1,120 1,198/53/403	10 6/0/7	582 454/45/346
	Rest16	13	1,708 1,657/101/749	29 23/1/20	587 611/44/204

Table 9: Dataset statistics for various tasks. #Cat refers to the number of aspect categories in the set. POS, NEU, and NEG denote the number of positive, neutral and negative quads or triplets respectively.

ACL 2023 Responsible NLP Checklist

A For every submission:

☑ A1. Did you describe the limitations of your work?
7

☑ A2. Did you discuss any potential risks of your work?
8

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

☒ A4. Have you used AI writing assistants when working on this paper?
Left blank.

B ☑ Did you use or create scientific artifacts?

3

☑ B1. Did you cite the creators of artifacts you used?
3, 5

☑ B2. Did you discuss the license or terms for use and / or distribution of any artifacts?
3, 8

☑ B3. Did you discuss if your use of existing artifact(s) was consistent with their intended use, provided that it was specified? For the artifacts you create, do you specify intended use and whether that is compatible with the original access conditions (in particular, derivatives of data accessed for research purposes should not be used outside of research contexts)?
8

☑ B4. Did you discuss the steps taken to check whether the data that was collected / used contains any information that names or uniquely identifies individual people or offensive content, and the steps taken to protect / anonymize it?
8

☑ B5. Did you provide documentation of the artifacts, e.g., coverage of domains, languages, and linguistic phenomena, demographic groups represented, etc.?
1, 5

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

C ☑ Did you run computational experiments?

3

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

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

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

☑ C3. Did you report descriptive statistics about your results (e.g., error bars around results, summary statistics from sets of experiments), and is it transparent whether you are reporting the max, mean, etc. or just a single run?
4

☑ C4. If you used existing packages (e.g., for preprocessing, for normalization, or for evaluation), did you report the implementation, model, and parameter settings used (e.g., NLTK, Spacy, ROUGE, etc.)?
3

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

Left blank.

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

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

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

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

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