|  |
| --- |
| **E2TPP: Improving Aspect-Based Sentiment Analysis via Element to Tuple Prompting Procedure** |
|  |
|  |

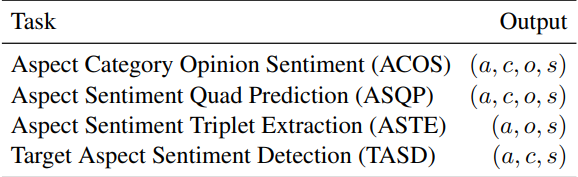
**Abstract**

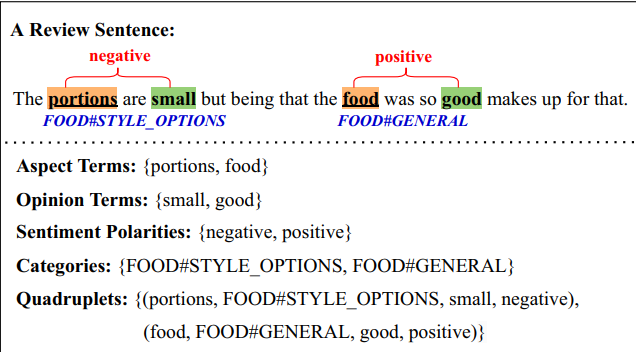
Aspect-based sentiment analysis (ABSA) has garnered increasing attention in recent times. Generative methods have significantly contributed to this task owing to their capability, flexibility, and the resulting enhancements. ABSA encompasses various tasks, depending on the targeted elements, and can be configured differently during training based on the data conditions. However, many existing studies typically predict target tuples simultaneously. In this work, we present the Element to Tuple Prompting Procedure (E2TPP), which introduces a two-step prompting architecture for predicting sentiment tuples. The first step focuses on predicting single elements, while the second step completes the process by mapping the predicted elements to their corresponding elements, yielding the final tuples. Specifically, E2TPP leverages the intuition of human-like problem-solving processes by breaking down a complex task into smaller parts, utilizing the first step's output as a guide in the second step. Beyond task-specific experiments, this paper addresses multi-task, low-resource, task transfer, and cross-domain scenarios, demonstrating its effectiveness and flexibility. Our proposed approach achieves new state-of-the-art results in almost all tasks and competitive results in others.

**1 Introduction**

Aspect-based sentiment analysis (ABSA), a nuanced sentiment analysis task, has garnered increased attention in recent years. ABSA seeks to forecast tuples of sentiment elements within a given text. ABSA research revolves around four key sentiment elements: aspect term (a), aspect category (c), opinion term (o), and sentiment polarity (s). Illustratively, considering the sentence "I had the Lamb special, which was perfect." the corresponding quadruplet (Lamb special, FOOD#QUALITY, perfect, positive)

Code and data released at https://github.com/mghiasvandm/E2TPP



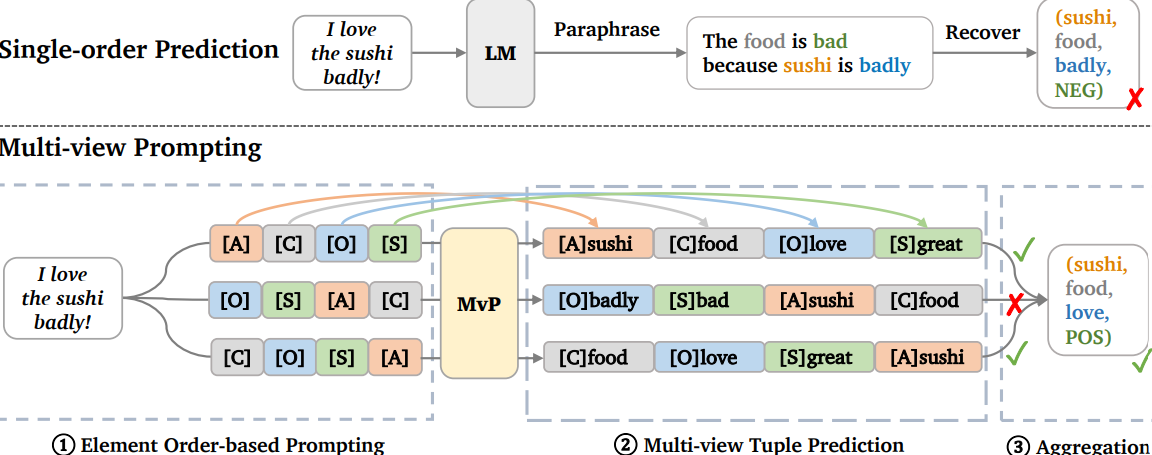


is produced, where "Lamb special" is the aspect term

categorized under the aspect category "FOOD#QUALITY," with the opinion term "perfect" conveying a positive sentiment polarity.

Correspondingly, ABSA has been decomposed into multiple subtasks based on predicting different target elements. Primitive works have studied the extraction of a single term, such as aspect term or opinion term extraction, or aspect category or sentiment polarity separately. They have also explored double terms, such as the extraction of aspect-opinion pairs, category-sentiment pairs, and aspect-sentiment pairs separately. However, recent studies have focused on the extraction of triplets and quadruplets that are grouped into four items: aspect sentiment triplet extraction (ASTE), target aspect sentiment detection (TASD), aspect sentiment quad prediction (ASQP), and aspect category opinion sentiment (ACOS). Their target formats are shown in Table 1.

Typically, ABSA approaches can be divided into two components: generative and machine reading comprehension. Recently, machine reading-based methods have utilized encoder-only models and predicted sentiment tuples by categorizing ABSA elements into two groups: extraction and classification. Aspect and opinion terms are in the extraction group, while category and sentiment are predicted as a classification task. In addition to the previous statements, the most recent approach, EMRC, has applied a bi-directional attention mechanism to enhance the strong association between



the context and queries, making the context representation task-aware. While any ABSA task can be converted into a generative problem, various generative approaches have been proposed for the mentioned tasks. The common practice has been to generate a sequence of sentiment elements in a specified format to leverage label semantics. Specifically, 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), multi-view state (Gou et al., 2023) as the target of the generation models.

Beyond task-specific settings, other configurations such as cross-domain, task transfer, multi-task, and low-resource settings have recently garnered significant attention. LEGO-ABSA and MVP have explored task transfer and multi-task scenarios for the first time. Additionally, MVP has experimented with a low-resource setting for this task and generalized its strategy to this scenario. Despite their success in supervised in-domain settings, their effectiveness has yet to be validated in the cross-domain setting. The BGCA paper introduces a novel, simple, yet effective target data augmentation strategy and has achieved state-of-the-art performance with this approach.

However, most previous works usually try to generate the target sequence at once and in one step, neglecting the potential benefits of using a two-step approach. This approach divides the problem into simpler subtasks, which is a technique inspired by the human-like problem-solving process. Additionally, the approach presented in the LEGO-ABSA paper is similar to a multistep framework, but based on the conflicts observed during its inference, its strategy does not seem promising.

To address the above issue and intuition, we propose the ***E****lement* ***to******T****uple* ***P****rompting* ***P****rocedure* (E2TPP), which uses a two-step architecture. In the

first step, it aims to predict some single elements as candidates for gold elements separately. In the second step, it maps and completes tuples associated with that element. In E2TPP, we have modeled ABSA tasks to pursue two goals: i) predicting the most possible single elements and ii) achieving accurate mapping and completion to predict tuples guided by the predicted elements. Our proposed E2TPP framework exhibits some unique advantages. Firstly, it effectively divides the problem into two simpler ones and trains separate expert models for each, where the improvement of each of the two subtasks will have a clear effect on the final outputs. Secondly, as E2TPP generates different element types in the first step, it is able to form target tuples via different paths. Through a final aggregation, it selects the most probable tuples. Forming target tuples via different paths and starting points, and this type of recovery can be helpful when predicting specific element types from the inputs seems to be complex.

In this paper, we present two modules that contribute to the mentioned strategy, named single-view E2TPP (SVE2TPP) and multi-view E2TPP (MVE2TPP), both of which are thoroughly explained in the methodology section.

In summary, our major contributions are as follows:

1) We introduce E2TPP, a simple yet effective two-step prompting framework with two different modules aimed at simplifying all ABSA tasks. This framework improves ABSA final predictions by following the corresponding procedure and is applicable to other NLP tasks whose output texts can be modeled as a tuple of elements.

2) By integrating the BGCA data augmentation strategy with E2TPP, we have enhanced ASTE in the cross-domain configuration, which has posed a recent and challenging problem.

3) E2TPP naturally allows us to train a single model simultaneously on all tasks.

4) We have successfully implemented E2TPP in low-resource settings and task transfer scenarios, distilling the strategy's capabilities.

5) Experiments show that E2TPP significantly advances the state-of-the-art and even ChatGPT (gpt-3.5-turbo) in both zero-shot and few-shot on most datasets in different settings.

**2 Methodology**

**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).

**2.2 Training**

To control the prediction order of sentiment elements, E2TPP introduces a two-step prompting mechanism. Specifically, we design the target and input separately for each step and fine-tune a T5-base model separately for each of them. Despite the inference step, in the training step, the models work independently and separately from each other with their own data. Both of the models are fine tuned with AdamW optimizer and with the Cross-Entropy loss function.

**2.2.1 First Step Input and Target Schema**

If we consider the input as “The portions are small, but being that the food was so good makes up for that.” as X and we have initialized the ASQP task, for instance. In the first step, we try to predict all available single elements, which are of one type from the aspect, category, opinion, or sentiment types, by constructing the input as follows:

Input = X => aspects: [A]

Target = [A] portions, food

Input = X => categories: [C]

Target = [C] FOOD#STYLE\_OPTIONS, FOOD#GENERAL

Input = X => opinions: [O]

Target = [O] small, good

Input = X => sentiments: [S]

Target = [S] negative, positive

First, we provide all of these four prompts for each data point in the dataset, and then fine-tune the T5-base model on this seq2seq task.

**2.2.2 Second Step Input and Target Schema**

If we consider the sentence as "The pizza is delicious and the salads are fantastic." and have initialized the ASQP task, for instance. In the second step, we attempt to predict and complete the tuples associated with the extracting elements in the first step by correctly mapping them via the existence of the element as a guide in the input. In the training step, elements are treated as completely equal to gold elements. First, for each datum, we have two sequences as input and output, as shown below:

Input: The pizza is delicious and the salads are fantastic.

Target: (pizza, FOOD#QUALITY, delicious, positive); (salads, FOOD#QUALITY, fantastic, positive)

Next, we consider the mentioned input as X and group the target tuple elements based on their types:

Aspects: pizza, salads

Categories: FOOD#QUALITY, FOOD#QUALITY

Opinions: delicious, fantastic

Sentiments: positive, positive

For writing simplification, we have:

a1 = pizza & a2 = salads

c = FOOD#QUALITY

o1 = delicious & o2 = fantastic

s = positive

To construct the input data of the second step, if the main input sentence is X, for this data instance, new inputs and targets are produced as follows:

Aspect Group:

Input = X -> a1: aspect, category, opinion, sentiment

Target = a1, c, o1, s

Input = X -> a2: aspect, category, opinion, sentiment

Target = a2, c, o2, s

Category Group:

Input = X => c: category, aspect, opinion, sentiment

Target = c, a1, o1, s; c, a2, o2, s

Opinion Group:

Input = X -> o1: opinion, sentiment, aspect, category

Target = o1, s, a1, c

Input = X -> o2: opinion, sentiment, aspect, category

Target = o2, s, a2, c

Sentiment Group:

Input = X => s: sentiment, opinion, category, aspect

Target = s, o1, c, a1; s, o2, c, a2

Some points are worth to mention:

* Using “=>” instead of “->” in the input of this step occurs while more than one tuple is associated with that specific element; This item has been set to make the model sensitive to provide more than one tuple while seeing “=>” in the input.
* The mentioned (above) explanation of the for of data for the second step is for the first module of this project which is a single view type. But about the other module, it is needed to mention that the second module will be created while applying multi-view prompting to the data in second step. It means for each of the groups in the second step fixing the first prompt element and gathering data for all of the permutations of the other prompting elements, for example in aspects groups, we keep the “aspect” word fixed in the input as the prompt but change the order of terms “category”, “opinion” and “sentiment” besides changing the order of elements in the target sequence, too for applying a data augmentation approach which is studied in MVP paper.
* For the second step, despite of previous papers which used masking templates for their model prediction, in this work we propose an extraction style rather than the masking style according to the following reasons: i) To leverage element types semantics, instead of using mask tokens such as “[A]”, “[O]”, etc we prefer to use the main name of elements such as “aspect”, “opinion”, etc. ii) By selecting masking style the target sequences length will be much more longer which increases the training time and the need to more GPU resources. iii) After doing multiple experiments, we’ve understood that despite of its simplicity, extraction style performs much better than the masking style.

Finally, we provide all of the corresponded prompts for each data point in the dataset, and then fine-tune the T5-base model on this seq2seq task.

**2.3 Inference**

In the batch inference stage, unlike training, the two models perform dependently, first we feed the step-one model the test data and it generates the possible elements; Then, similar to the training process for the second step, we group the predicted elements and construct all of the required prompts for the second step and finally generate the outputs. During inference two steps of constrained beam search and aggregation step are crucial.

**2.3.1 Constrained Beam Search**

Code not ready yet

**2.3.2 Aggregation of Final Outputs**

After the generation of final results from the second model, we need to aggregate results and select the final tuples as our answer for each data point; By predefining m as the number of prompts of a specific data point, and T as a set of the union of all of the predicted tuples for that data point and the function n(x, y) find the frequency of tuple y inside tuples set T, then final aggregated results are calculated based on formula, below: and if the set is empty, the m/2 threshold is lowered to (m/2)-1 and maybe also (m/2)-2.

**3** **Related Works**

**Aspect-based 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.** 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. And as the most generative methods we can refer to TAGS paper and MVP paper which have used a sequence tagging and data augmentation approach for solving the task which MVP has done this by utilizing different orders of prompting elements while training and inference. Despite the promising results, most of the current methods either lack from a two-step effective strategy which divides the problem into simpler ones or having conflict while inference. Our method has utilized this opportunity and with a simple yet effective strategy the modelled version of this task suits the main task well.

**Cross-domain ABSA.**

Cross-domain ABSA aims to utilize labeled data from a source domain to gain knowledge that can be applied to a target domain where only unlabeled data is available. The main research line of cross-domain ABSA involves two paradigms: feature-based adaptation and data-based adaptation (Zhang et al., 2022). Feature-based adaptation focus on learning domain invariant features. Some have utilized domain independent syntactic rules to minimize domain gap (Jakob and Gurevych, 2010; Chernyshevich, 2014; Ding et al., 2017; Wang and Pan, 2018, 2019), while others have employed domain discriminators to encourage the learning of universal features (Li et al., 2019c; Yang et al., 2021; Zhou et al., 2021; Zhang et al., 2021a). On the other hand, data-based adaptation aims to adapt the training data distribution to the target domain. They either adjust the importance of individual training instances through re-weighting (Xia et al., 2014; Gong et al., 2020), or generate additional training data using another pre-trained model (Yu et al., 2021; Li et al., 2022). Despite their effectiveness, most of these works require task-specific design or external resources, preventing easy extensions to other cross-domain ABSA tasks. The SOTA paper in this task is BGCA which performs a data augmentation algorithm for distilling target data knowledge while training; According to the flexibility of E2TPP and also the robust ability of BGCA, by mixing these two strategies, we’re a able to enhance the distilling target knowledge ability in the source model.