SentiPrompt: Sentiment Knowledge Enhanced Prompt-Tuning for Aspect-Based Sentiment Analysis

Chengxi Li\textsuperscript{1,2}, Feiyu Gao\textsuperscript{2}, Jiajun Bu\textsuperscript{1,3}, Lu Xu\textsuperscript{2}, Xiang Chen\textsuperscript{1}, Yu Gu\textsuperscript{1}, Zirui Shao\textsuperscript{1}, Qi Zheng\textsuperscript{2}, Ningyu Zhang\textsuperscript{1,3}, Yongpan Wang\textsuperscript{2}, Zhi Yu\textsuperscript{1,3*}

\textsuperscript{1}Alibaba-Zhejiang University Joint Institute of Frontier Technologies, Hangzhou, China  
\{chengxili, bj.jiang.chen, zhangningyu, yuzhirenzhe\}@zju.edu.cn  
\textsuperscript{2}Damo Research, Alibaba Group  
\{feiyu.gfy.lu.x.yongqi.zq\}@alibaba-inc.com  
\textsuperscript{3}Ningbo Research Institute, Zhejiang University, Ningbo, China

Abstract

Aspect-based sentiment analysis (ABSA) is an emerging fine-grained sentiment analysis task that aims to extract aspects, classify corresponding sentiment polarities and find opinions as the causes of sentiment. The latest research tends to solve the ABSA task in a unified way with end-to-end frameworks. Yet, these frameworks get fine-tuned from downstream tasks without any task-adaptive modification. Specifically, they do not use task-related knowledge well or explicitly model relations between aspect and opinion terms, hindering them from better performance. In this paper, we propose SentiPrompt to use sentiment knowledge enhanced prompts to tune the language model in the unified framework. We inject sentiment knowledge regarding aspects, opinions, and polarities into prompt and explicitly model term relations via constructing consistency and polarity judgment templates from the ground truth triplets. Experimental results demonstrate that our approach can outperform strong baselines on Triplet Extraction, Pair Extraction, and Aspect Term Extraction with Sentiment Classification by a notable margin.

Introduction

Aspect-based Sentiment Analysis (Pontiki et al. 2014) is a fine-grained sentiment analysis task which requires extracting aspects, judging the corresponding polarities, and finding opinions as the causes of sentiment towards each aspect. For the example in Figure 1, the task aims to extract the two aspects “owners” and “beer selection”, their corresponding opinions (“great fun” and “worth staying for”) and polarities (both positive). Previous studies have proposed many subtasks for ABSA, all being different in required input, output and task formulations. In this paper, we focus on the most complex and challenging three subtasks, ASEC, Pair, and Triplet, as shown in Figure 2. Aspect Term Extraction and Sentiment Classification (ASEC) requires extracting the aspect terms and classifying the sentiment polarities about them. Pair Extraction (Pair) extracts the aspect terms as well as the corresponding opinion terms simultaneously. Aspect Sentiment Triplet Extraction (Triplet) (Peng et al. 2020) aims to get the information containing all the aspects, their corresponding opinions and sentiment polarities from input sentences. These are all nearly complete solutions for the ABSA task, especially Triplet.

Researchers have proposed pipeline approaches (Peng et al. 2020; Mao et al. 2021) to solve these subtasks in a multi-stage framework. However, these multi-stage models are not end-to-end, thus, making predictions from various sub-models inside separately. Separating tasks into multiple stages potentially breaks the relation modeling within the triplets and brings about ineligible error propagation. Recently, several neural-network-based models (Wu et al. 2020a; Xu et al. 2020; Xu, Chia, and Bing 2021) have been proposed to develop end-to-end framework with sequence tagging. In these approaches, aspects and opinions can be jointly extracted and polarities are also jointly considered. However, these approaches mainly rely on modeling the relations between aspects and opinions at the word level, regardless of the span-level relations which need supervisions from extra subtasks for pruning. More recent studies make attempts on unified frameworks with Pre-trained Language Models (PLM) for the ABSA task (Chen et al. 2021a; Yan et al. 2021), yielding promising performance on benchmark datasets. Despite the success of PLM fine-tuning, some recent studies find that one of its critical challenges is the significant gap of objective forms between pre-training and fine-tuning, which restricts PLMs from reaching their full potential.

Prompt-tuning (Jiang et al. 2020; Shin et al. 2020; Schick and Schutze 2021; Han et al. 2021), as a new fine-tuning...
paradigm, has emerged which is proposed to bridge the gap of objective forms between pre-training and fine-tuning. With appropriate prompts and tuning objectives, the prompt-tuning could manipulate the model behavior to fit various downstream tasks. By using specially constructed prompts, we can further inject and stimulate the task-related knowledge in PLMs, thus boosting the model performance. Nevertheless, handcrafting the appropriate prompt for ABSA requires domain expertise, and auto-constructing a well-performing prompt often requires additional computational cost for verification.

Motivated by the above observations, in this paper, we propose SentiPrompt, a novel approach to leverage sentiment knowledge enhanced prompts to tune the LM in a generative framework for ABSA. To be specific, we leverage consistency and polarity judgment templates to construct prompts regarding ground truth aspects, opinions, and polarities as sentiment knowledge. During training, the model predicts the consistency of prompt samples with the ground truth and classify polarities for consistent ones as instructed. In this way, the encoder part of the framework learn to explicitly capture sentiment relations between aspect and opinion term pairs. Similarly, we introduce sentiment knowledge by this tuning process. Together with the supervision from generative predictions, the framework can serve the ABSA task far better than solely being fine-tuned by the loss from final predictions. This is the first attempt to introduce downstream task-related knowledge and to enhance relation modeling for the LM through prompt-tuning methods. Sentiment knowledge enhanced prompt-tuning, working as an ABSA task-adaptive modification to the tuning process, helps to promote term extraction, pairing and classification with sentiment knowledge enhanced prompts. This technique needs no extra parser to get semantic information or building graph from sentence structure. Besides, neither aspect and opinion terms nor polarities are of large numbers in a comment. Thus it allows convenient, time-sparing and explicit modeling of relations between those terms and polarities and better introduction of sentiment-related knowledge for the LM.

Our main contributions can be summarized as follows:

- We propose sentiment knowledge enhanced prompt-tuning (SentiPrompt) to explicitly model the relations between terms and leverage sentiment knowledge. We further utilize trainable continuous templates to obtain optimized prompts. To the best of our knowledge, this is the first approach to incorporate prompt-tuning for ABSA subtasks.
- We revise the latest version of four benchmark datasets regarding mistakes and incompleteness in triplet annotations and release a new version for future research. Experimental results demonstrate that our method archives promising improvements and obtains new state-of-the-art performance all datasets.

Related Works

Aspect-based Sentiment Analysis

Early research about the ABSA task starts with extracting aspects or opinions alone. Most early research follows sequence tagging schema. Li et al. (2018), Xu et al. (2020), Bi et al. (2021) to solve Aspect Term Extraction (ATE) while recent works Ma et al. (2019), Li et al. (2020) attempt to use sequence-to-sequence techniques with PLMs and achieve relatively promising results. As for Opinion Term Extraction (OTE), most works treat it as an auxiliary task (Wang and Pan, 2018; He et al. 2019; Chen and Qian, 2020). Later research turns to extract the corresponding opinions (Aspect-oriented Opinion Extraction, AOE) or classifying polarities (Aspect-level Sentiment Classification, ALSC) for every given aspect. Fan et al. (2019) first introduces AOE with their baseline model and datasets. Lately, others (Wu et al. 2020b; Pouran Ben Veyseh et al. 2020) develop some sequence tagging models to follow this track. For ALSC, works about it start earlier. Most of them Tang et al. (2016), Wang et al. (2016) incorporate long-short-term-memory (LSTM) units or attention mechanism to deal with relations between aspects and sentiments. In addition, some works (Chen, Teng, and Zhang 2020; Wang et al. 2020; Tang et al. 2020; Li et al. 2021) propose to leverage graph neural networks to utilize syntax and semantic information from extra parsers.

Recent studies propose some more fine-grained subtasks: AESC, Pair, and Triplet for ABSA. Peng et al. (2020) first proposes a multi-stage pipeline to do extraction, pairing and classification step by step. Then Xu et al. (2020) follows the commonly used sequence tagging schema with a position-aware method. Yet another line (Wu et al. 2020a) tries to solve Triplet by labeling the relations of word-pairs. Xu, Chia, and Bing (2021) goes further by labeling span-pairs with a pruning method. Moreover, Mao et al. (2021), Yan et al. (2021) propose to combine a set of subtasks through unified formulations. They formulate various subtasks as a machine reading comprehension or generation task, which perform quite well. Our method tends to solve this task in a unified generative framework yet include explicit modeling of relations between terms through sentiment knowledge enhanced prompt-tuning.

Prompt-tuning

Prompt-tuning is a new paradigm of fine-tuning inspired by GPT-3 (Brown et al. 2020), especially for language models in few-shot or zero-shot settings. It means prepending instructions and demonstrations to the input and output predictions. Recent PET work (Schick and Schütze, 2021) focus on semi-supervised setting with a large number of unlabeled examples. Gao, Fisch, and Chen (2020) explore their prompt-tuning methods with demonstrations on language models for some benchmark tasks, including Sentiment Classification. Prompt-tuning can induce better performances for PLMs on widespread of NLP tasks including text classification (Gao, Fisch, and Chen, 2020), relation extraction (Han et al. 2021, Chen et al. 2021b), NER (Chen et al. 2021c), entity typing (Ding et al. 2021), and so on. To construct better prompt for downstream tasks, several approaches Han et al. (2021), Hu et al. (2021) leverage knowledge injection (Zhang et al. 2021a,b) to templates and verbalizer construction. Besides, there exist lots of works on prompting for mining knowledge from PLMs (Davison, Feldman, and Rush, 2019; Petroni et al. 2019; Talmon et al. 2020).
Unlike those works, we focus on ABSA which is quite different from classification tasks. Furthermore, since handcrafting a well-performing prompt is rather difficult, a number of works focus on automatic prompt search via generation or gradient-based search (Jiang et al. 2020; Shin et al. 2020; Gao, Fisch, and Chen 2020). Those works mentioned above still keep the discrete format for prompt search whereas [Liu et al. 2021; Li and Liang 2021] tends to do optimization for continuous prompt embeddings.

**Methodology**

In this section, we first present the task formulation and the sequential output generated to cover all subtasks, then elaborate our proposed method about sentiment knowledge enhanced prompt-tuning and the generation module.

**Task Formulation**

For a given sentence $X = [x_1, x_2, \ldots, x_n]$, the ABSA task aims to provide the aspect terms, the sentiment polarity, and the opinion terms, represented by $a$, $s$ and $o$ in our method respectively. $a$ and $o$ are span indexes of terms and $s \in \{POS, NEG, NEU\}$ is the polarity class index. Superscripts $\text{start}$ and $\text{end}$ denote the start and end index of the corresponding token span in the sequence. For our generative framework, the target sequence $Y = [y_1, y_2, \ldots, y_m]$ consists of multiple base prediction $\text{pred} = [a_{\text{start}}, a_{\text{end}}, o_{\text{start}}, o_{\text{end}}, s]$. Base predictions of subtasks are all some subsets of the above $\text{pred}$ and we list them as follows:

- For AESC: $\text{pred} = [a_{\text{start}}, a_{\text{end}}, s]$
- For Pair: $\text{pred} = [a_{\text{start}}, a_{\text{end}}, o_{\text{start}}, o_{\text{end}}]$
- For Triplet: $\text{pred} = [a_{\text{start}}, a_{\text{end}}, o_{\text{start}}, o_{\text{end}}, s]$

In our work, we use BART (Lewis et al. 2020) with pointer network (Vinyals, Fortunato, and Jaitly 2015) for sequence-to-sequence generation. During training, the input $X$ is used for both sentiment knowledge enhanced prompt-tuning and index sequence generation. The encoder part is shared across both sides.

**SentiPrompt Tuning**

The sentiment knowledge of ground truth sentiment terms and polarities is injected in the construction of prompts. The prompt encoder is inserted here to provide trainable and continuous representations for prompts.

**Sentiment Knowledge Enhanced Prompt Construction**

SentiPrompt tunes the LM with sentiment knowledge enhanced prompts as a masked language modeling task (MLM) during training. So we construct prompts as samples for SentiPrompt MLM based on terms and polarities from ground truth triplets. To construct a prompt masked training sample, we randomly take an aspect and an opinion from ground truth triplets to fill in a fixed template. Given an input sequence $X$ (e.g. \textit{Good Sushi High Price}), a template $T$ consists of pseudo prompt tokens $P_k$ and randomly sampled aspect and opinion terms $A$, $O$ (sampled from (\textit{Sushi}, \textit{Price}) and (\textit{Good}, \textit{High})). Here $P_k$ can be specific English words or just a randomly-initialized pseudo token to be optimized. It is formed like:

$$[P_{1:l_1}, A, P_{l_1+1:l_2}, O, [\text{MASK}]]$$  \hspace{1cm} (1)

Besides, we randomly do span manipulations to aspects and opinions of some prompts in construction. These prompts will be constructed with a sub-span of aspect and opinion terms if these terms contain more than one token. For single token cases, we attach a few neighbor tokens of the term to it. On the whole, the created prompts may be consistent with the ground truth triplets or against them due to random sample choices and random span manipulations. For the example input, the ground truth triplets are (\textit{Sushi}, \textit{Good}, \textit{POS}) and (\textit{Price}, \textit{High}, \textit{NEG}). Therefore, other possible pairs are all considered inconsistent. In Table 1, the first example is constructed from (\textit{Sushi}, \textit{High}) and the second by attaching the left neighbor \textit{Good} to the aspect \textit{Sushi}. For chosen pairs that are consistent with the ground truth, we will attach an extra polarity judgement prompt to it as the suffix. Likewise, the template $T'$ for the polarity judgement is formed like:

$$T' = [P_{l_2+1:l_p}, [\text{MASK}]]$$  \hspace{1cm} (2)

The label ground truth of the very polarity judgement prompt is the same as the corresponding polarity of the original source triplet. For negative pairs, because of the mismatch of aspect and opinion terms, judging the polarity of such a pair is meaningless. Thus, in this case, only the aspect-opinion combination part will be fed into the model. The whole template $T$ for prompt is like:

$$T = [P_{1:l_1}, A, P_{l_1+1:l_2}, O, [\text{MASK}]]$$  \hspace{1cm} (3)

In this way, we can formulate a simple classification task for SentiPrompt MLM through a prompt $X^p$ according to the input sentence $X$ with the extra polarity judgement prompt $X'^p$ as the suffix. Take a manually designed one as an example:

$$X^p = \text{The \ is } O? \ [\text{MASK}], X'^p$$  \hspace{1cm} (4)

$$X'^p = \text{This is } [\text{MASK}]$$  \hspace{1cm} (5)

We use this as an auxiliary task for the encoder part to decide whether to fill in “yes” (consistent with ground truth) or “no” (inconsistent) for the masked position. In addition, if the polarity judgement prompt is attached, let the encoder part decide the right polarity class to fill in that masked position. Let $y^p \in S = \{“yes”, “no”, POS, NEG, NEU\}$ be the polarity class token label for [MASK], the probability of predicting consistency is:

$$p(y^p|X, X^p) = p([\text{MASK}] = y^p|X, X^p)$$

$$= \frac{\exp(w_{y^p} \cdot h[\text{MASK}])}{\sum_{y^p \in S} \exp(w_{y^p} \cdot h[\text{MASK}]})$$  \hspace{1cm} (6)

where $h[\text{MASK}]$ is the hidden vector of the [MASK] position and $w$ means the weight vector before softmax for the word. For this sentiment knowledge enhanced prompt MLM task, we compute the cross-entropy loss of the predictions on masked tokens:

$$L_{\text{prompt}} = - \sum \sum y^p \log(p(y^p|X, X^p))$$  \hspace{1cm} (7)
Prompt Encoder For better prompt representations, we replace the embeddings of pseudo prompt tokens $P_k$ from BART with their differential output embeddings through a prompt encoder and optimize them for the downstream task. To associate these pseudo prompt tokens with each other, we choose a bidirectional long-short term memory networks to connect these separate token embeddings and two layers of multi-layer perceptron activated by ReLU to get the processed hidden states. The prompt encoder takes the embeddings of $l_p$ pseudo prompt tokens $P_k$ from a prompt-token embedding layer $e_p$ as input and outputs the optimized embedded $h'_k$.

$$ E^p = e_p(P) $$

$$ h'_k = \text{MLP}(\text{LSTM}(h_{0:k-1}); \text{LSTM}(h_{k:l_p})) $$

where $h'_k \in \mathcal{R}^d$. The optimized embedding at each position of pseudo prompt tokens will replace the embedding from the layer $e$ of BART in SentiPrompt MLM. So the tokens in $T$ will be mapped as:

$$ [h'_{1:k+t}, e(A), h'_{t+1:k+t}, e(O), e([\text{MASK}])] $$

and then be fed into the encoder part while tokens of $A$, $O$ and $[\text{MASK}]$ will still be embedded by the BART embedding layer $e$.

Prompt Optimization for ABSA Given a PLM $\mathcal{M}$ and an input sequence $X$, the pre-trained embedding layer $e \in \mathcal{M}$ will map every token $[x_i^m]_{i=1}^n$ of $X$ to input embeddings $e(x_i^m)$. Let $\mathcal{V}_\mathcal{M}$ be the vocabulary of $\mathcal{M}$, the prompt tokens $P_k$ in $T$ and $T'$ satisfy that $P_k \in \mathcal{V}_\mathcal{M}$ as well as the label words $S$ for $[\text{MASK}]$. As $h'$ is from prompt encoders with trainable parameters, this allows better continuous prompts beyond the original vocabulary $\mathcal{V}_\mathcal{M}$ could express. With the downstream loss, these sentiment knowledge enhanced prompts can be differentially optimized by

$$ \hat{h}'_{1:l_p} = \arg \min h' \mathcal{L}(\mathcal{M}(X, X^p, Y, Y^p)) $$

Generation Framework for ABSA

For the ABSA task, the model takes a sequence of tokens $X$ as input and hope to generate the target sequence $Y$ as defined above. The input and output sequence starts and ends with special tokens: $<s>$ and $<\backslash s>$. These special tokens are to be generated in $Y$ but we ignore them in equations for simplicity. Given a sequence of tokens $X$:

$$ P(Y|X) = \prod_{t=1}^m p(y_t|X, y_1, y_2, ..., y_{t-1}) $$

To model the index probability $p_t$ for every $t$, we use BART model with pointer network, which roughly consists of an Encoder part and a Decoder part.

Encoder The encoder part is to encode $X$ into the hidden representation space as a vector $H^e$.

$$ E^e = e(X) = [e(x_1), e(x_2) ... e(x_n)] $$

$$ H^e = \text{BARTEncoder}(E^e) $$

where $E^e, H^e \in \mathcal{R}^{n \times d}$, $d$ is the hidden state dimension and $e$ is the embedding layer.
Decoder The decoder part takes the encoder outputs $H^e$ and previous decoder outputs $y_1, y_2, \ldots, y_{t-1}$ as inputs and decode $y_t$ out. As $[y_{j=1}^{t-1}]$ are token indexes, an Index-to-Token Converter is applied to make a conversion.

$$y_j = \begin{cases} X_{y_j}, & \text{if } y_j \text{ is a pointer index} \\ C_{y_j-n}, & \text{if } y_j \text{ is a polarity class index} \end{cases}$$  \tag{15}

where $C = [c_1, c_2, \ldots, c_l]$ is the polarity class token list.\footnote{The polarity class token always starts after the pointer indexes of the given sequence, at $n + 1$.}

After this, we use a BART decoder to encode tokens to get the decoder hidden state for $y_t$.

$$h_t = \text{BARTDecoder}(H^e; [\hat{y}_{j=1}^{t-1}])$$  \tag{16}

where $h_t \in \mathcal{R}^d$, and the probability distribution $p_t$ of token $y_t$ can be computed as:

$$\tilde{H}^e = \alpha \cdot \text{MLP}(H^e) + (1 - \alpha) \cdot E^e$$  \tag{17}

$$C^e = e(C)$$  \tag{18}

$$p_t = \text{Softmax}([\tilde{H}^e; C^e] \cdot h_t)$$  \tag{19}

where $\tilde{H}^e \in \mathcal{R}^{n \times d}$, $C \in \mathcal{R}^{l \times d}$ and $p_t \in \mathcal{R}^{(n+l)}$ is the predicted probability distribution of $y_t$ on all candidate indexes.

Training and Inference
During training, we compute the weighted sum of the cross-entropy losses from sentiment knowledge enhanced prompt-tuning and index predictions for optimization of the generation framework and continuous prompt representations.

$$\mathcal{L}(\theta) = \alpha_1 \cdot \mathcal{L}_{\text{prompt}} + \alpha_2 \cdot \mathcal{L}_{\text{gen}}$$ \tag{20}

During inference, we only use $X$ for generation. We apply beam search to generate the prediction sequence in an autoregressive way. Then, a specific decoding process is applied to convert the output sequence into required terms and polarities. The corresponding decoding algorithm is included in Appendix G.

Experiments
In this section, we first introduce the datasets on which the experiments are conducted and detail those baselines to be compared with, then present our main results.

Datasets
We evaluate our method on three versions of four ABSA datasets. The first version ($D_{20a}$) is from Peng et al. (2020). They get the paired annotated opinions from Wang et al. (2017), the corresponding aspects annotation from Fan et al. (2019), and refine the data into $<$a, o, s$>$ triplet form. The second version $D_{20b}$ (Xu et al. 2020) is a revised version of $D_{20a}$. The third version $D_{20c}$ (AESC) are included in Appendix D and E. We will release $D_{21}$ later.

Baselines
To put our results in perspective, we summarize state-of-the-art models on three ABSA subtasks and compare SentiPrompt with them. Detailed comparisons of baselines are shown in Appendix F. Those three with the suffix “∗” denote the variant version by Peng et al. (2020) for being capable of doing AESC, Pair, and Triplet. They are modified by adding an MLP to determine whether a triplet is correct in the matching stage. The baseline results marked with $\dagger$ are from Xu et al. (2020) and $\ddagger$ are from Yan et al. (2021). The “PE” in Table 2 indicates the prompt encoder. More details about implementations and hyper-parameter settings are shown in Appendix A.

Main Results
On $D_{20a}$, we compare our method on AESC, Pair and Triplet. Experimental results on $D_{20a}$ are shown in Table 2. Our method obtains convincing improvements on all sub-tasks of all datasets, 3.38% F1 for AESC, 2.09% F1 for Pair and 3.60% F1 for Triplet on average. Experimental results for Triplet on $D_{20b}$ are shown in Table 3. According to the table, our method achieves better performance than baselines on all datasets. Our method surpasses strong baselines by 7.61% at most, with an average of 3.44% F1 for Triplet on all datasets. On the coming up $D_{21}$, we compare our method on Triplet in Table 3. SentiPrompt outperforms representative baselines by an average of 2.30% F1.
Table 2: Comparison of F1 scores for AESC, Pair, and Triple on the $D_{20a}$. We highlight the best results in bold.

| Model            | 14res | 14lap | 15res | 16res |
|------------------|-------|-------|-------|-------|
|                  | AESC  | Pair  | AESC  | Pair  | AESC  | Pair  | AESC  | Pair  |
| CMLA+ †          | 70.62 | 48.95 | 43.12 | 56.90 | 44.10 | 32.90 | 53.60 | 44.60 |
| RINANTE+ †       | 48.15 | 46.29 | 34.03 | 36.70 | 29.70 | 20.0  | 41.30 | 35.40 |
| Li-unified+ †    | 73.79 | 55.34 | 51.68 | 63.38 | 52.56 | 42.47 | 64.95 | 56.85 |
| Peng-two-stage † | 74.19 | 56.10 | 51.89 | 62.34 | 53.85 | 43.50 | 65.79 | 56.23 |
| JET-BERT ‡       | -     | -     | 63.92 | -     | -     | 50.0  | -     | -     |
| OTE-MTL ‡        | 45.05 | -     | 59.67 | -     | -     | 52.95 | -     | -     |
| Dual-MRC ‡       | 76.57 | 74.93 | 70.32 | 64.59 | 63.37 | 55.58 | 65.14 | 64.97 |
| SPAN-BART ‡      | 78.47 | 77.68 | 72.46 | 68.17 | 66.11 | 57.59 | 69.95 | 67.98 |

SentiPrompt        81.09 78.96 75.01 70.79 68.05 60.41 74.20 70.36 64.50 79.81 80.13 74.66
SentiPrompt w/o PE 79.44 78.30 73.70 68.10 68.75 58.75 70.63 67.62 61.63 77.59 78.21 72.69

Table 3: Comparison of F1, Precision and Recall for Triple on the $D_{20b}$. We highlight the best results in bold.

| Model            | 14res | 14lap | 15res | 16res |
|------------------|-------|-------|-------|-------|
|                  |       |       |       |       |
| CMLA+ †          | 39.18 | 47.13 | 42.79 | 30.09 | 36.92 | 33.16 | 34.56 | 39.84 |
| RINANTE+ †       | 31.42 | 39.38 | 34.95 | 21.71 | 18.66 | 20.07 | 29.88 | 30.06 |
| Li-unified+ †    | 41.04 | 67.35 | 51.0  | 40.56 | 44.28 | 42.34 | 44.72 | 51.39 |
| Peng-two-stage † | 43.24 | 63.66 | 51.46 | 37.38 | 50.38 | 42.87 | 48.07 | 57.51 |
| JET-BERT ‡       | 70.56 | 55.94 | 62.40 | 55.39 | 47.33 | 51.04 | 64.45 | 51.96 |
| GTS-BERT ‡       | 67.76 | 67.29 | 67.50 | 57.82 | 51.32 | 54.36 | 62.59 | 57.94 |
| SPAN-BART ‡      | 65.52 | 64.99 | 65.25 | 61.49 | 56.19 | 58.69 | 59.14 | 59.38 |

SentiPrompt        72.79 72.94 72.86 63.40 58.60 60.90 62.97 62.06 62.51 70.20 73.35 71.74
SentiPrompt w/o PE 73.65 67.20 70.38 60.42 57.86 59.91 60.61 61.24 60.92 67.90 71.21 69.52

Table 4: Comparison of F1, Precision and Recall for Triple on the $D_{21}$. We rerun the baseline models according to their own sources on $D_{21}$ and report their results. The best results are highlighted in bold.

| Model            | 14res | 14lap | 15res | 16res |
|------------------|-------|-------|-------|-------|
|                  |       |       |       |       |
| OTE-MTL          | 50.81 | 58.24 | 54.27 | 47.97 | 39.68 | 43.44 | 56.90 | 38.67 |
| JET-BERT         | 72.79 | 30.01 | 42.50 | 61.80 | 25.40 | 36.00 | 69.56 | 37.13 |
| GTS-BERT         | 61.09 | 60.62 | 60.85 | 56.48 | 49.82 | 52.94 | 58.39 | 55.08 |
| SPAN-BART        | 64.36 | 64.24 | 64.30 | 60.41 | 51.95 | 55.86 | 54.00 | 54.10 |

SentiPrompt        66.10 63.37 64.71 61.30 55.32 58.15 61.81 62.06 61.93 68.66 69.04 68.85
SentiPrompt w/o PE 65.65 63.95 64.79 59.88 53.72 56.64 55.84 58.79 57.28 66.79 67.91 67.35

on all datasets. However, compared to the results on $D_{20a}$, the best performances go down by 8.07%, 3.78%, 5.23%, and 2.99% F1 on 14res, 14lap, 15res, and 16res respectively. As is reflected by the significant drops of performances, our revised annotations are more complicated and complete thus set a higher demand for the capability of models. Additionally, SentiPrompt generally outperforms the model without prompt encoder, except for 14res of $D_{21}$. In conclusion, main experimental results demonstrate that SentiPrompt is a new state-of-the-art method for the ABSA task.

To illustrate the differences between models, some triplet prediction cases from some baselines and our SentiPrompt on a sample sentence are shown in Figure 4. For the given example, OTE-MTL could only make one triplet prediction out of three. GTS simply takes the token “too” following “margaritas” as its opinion term. SPAN-BART is able to give the right predictions on “food” and “margaritas” as well as our method but fails to identify “waitress”. Our method gives a larger span on the opinion for “waitress”, which is comparatively reasonable. Consider that the annotation span for this is actually not that complete. Generally speaking, SentiPrompt generates more reasonable and accurate outputs in this case. These indicate that our method is more helpful to extract correct terms and capture accurate relations between them, which benefit the polarity classification as well. These improvements surely lead to better model performance. Yet, while our method achieves significant improvements on 14res, 15res and 16res, the improvement on 14lap is relatively marginal.

Analysis

Tuning the LM in unified frameworks using prompting methods with sentiment knowledge shows exciting potentials, but also has its limitations. We conduct some extensional experiments to further discuss SentiPrompt.

Few-shot ABSA

To further evaluate the potential of our method, we conduct few-shot setting experiments with 10% of training data on four datasets. Due to the instability of
Sentence: the food was great, the margaritas too but the waitress was too busy being nice to her other larger party than to take better care of my friend and me.

| Triplets                  | GT | OTET | OTE-MTL | Span-BART | Ours |
|---------------------------|----|------|---------|-----------|------|
| (food, great, POS)        | ✓  | ✓    | ✓       | ✓         | ✓    |
| (margaritas, great, POS)  | ✓  | ✓    | ✓       | ✓         | ✓    |
| (waitress, too busy, NEG) | ✓  | ✓    | ✓       | ✓         | ✓    |
| (margaritas, too, POS)    | ✓  | ✓    | ✓       | ✓         | ✓    |
| (waitress, too busy being nice, NEG) | ✓ | ✓ | ✓ | ✓ | ✓ |

Figure 4: A case study on an input with multiple triplets. GT is the ground truth.

Tuning on small datasets, we measure average performance across 5 different randomly sampled $D_{train}$ and $D_{dev}$, using a fixed set of seeds $D_{seed}$. This gives a more robust measure of model performance and a better estimate of the variance. As previous works about these ABSA subtasks in few-shot setting are scarce, we still adopt some baselines mentioned above for comparison in few-shot setting experiments. The comparison of experimental results on $D_{20k}$ and $D_{21}$ are in Appendix B. These figures show that SentiPrompt achieves better performances than all baselines on 14lap and 14res and outperforms the majority class of baselines on 15res and 16res. However, our results suffer from high variance. In worst cases, the performance drops up to 10% F1 below the average. In addition, unlike some other prompt-tuning methods, SentiPrompt won’t perform well if most parameters of the LM are frozen, which increases the computational cost of tuning in this process.

Prompt Length Analysis Additionally, to investigate the effect of prompts with different lengths to the ABSA task, we conduct experiments with prompts of various lengths on $D_{21}$. For automatically optimized prompts, we simplify the template for this type and control the length by the numbers of pseudo prompt tokens at three fixed positions of the template. Let $N = l_1 = l_2 - l_1 = l_P - l_2$, pseudo prompt tokens at these positions are of the same number $N$. We compare prompts with $N = 1, 2, 3$. Manually designed ones are as shown in Table 6. The comparison figure is included in Appendix C. In most cases, continuously optimized prompts achieve comparable or higher performance than manually designed ones. According to the figure, it is clear that the number of pseudo prompt tokens in a template have notable influence on model performance as subtasks and datasets change. As more pseudo tokens provide more trainable parameters, they enhance the capability of the model yet require more data and time to be properly trained.

Multi-triplet Analysis Besides, we compare the model performance of SPAN-BART and SentiPrompt on $D_{21}$ for multi-triplet cases, with more than one aspect or opinion within the input. These data account for about 70% of the total. As shown in Table 6, SentiPrompt improves the Precision quite a lot while keeps the Recall changing little, resulting in the general lift on F1. Consider the large proportion of multi-triplet cases in data, the improvements in main results can be largely attributed to better performance on them. With the LM capturing relations between multiple aspects and opinions better, SentiPrompt performs better on multi-triplet cases and thus benefits the overall performance.

| Model        | Metric | 14res | 14lap | 15res | 16res |
|--------------|--------|-------|-------|-------|-------|
| SPAN-BART    | P      | 66.27 | 63.07 | 58.29 | 59.14 |
|              | F1     | 59.35 | 48.76 | 48.05 | 71.08 |
| SentiPrompt  | P      | 62.62 | 55.00 | 52.68 | 64.56 |
|              | F1     | 70.13 | 68.09 | 64.48 | 77.03 |

Table 5: Extensional comparison of results for multi-triplet cases on $D_{21}$ dataset.

Invalid Prediction Analysis Since those subtasks require formatted sequence output to decode as described in Section 3, we estimate the invalid prediction rates for Triplet of the generative framework SPAN-BART and SentiPrompt, on error length (Err-length) and error order (Err-order), respectively. A valid triplet prediction $p_i$ should contain five indexes. The start index should be smaller or equal to the paired end index. From Table 6, SentiPrompt could achieve sufficiently low invalid rates for both error cases on most datasets. Lower invalid rate on 14res, 15res and error length decrease much on 16res. Lower invalid rates compared to SPAN-BART demonstrates that a better way of introducing knowledge for unified frameworks leads to well-formatted index sequence predictions, which lay a solid foundation for better actual performance in practical scenarios.

| Model        | Error (%) | 14res | 14lap | 15res | 16res |
|--------------|-----------|-------|-------|-------|-------|
| SPAN-BART    | Err-length| 0.48  | 0.77  | 1.41  | 1.40  |
|              | Err-order | 1.75  | 3.70  | 3.26  | 3.26  |
| SentiPrompt  | Err-length| 0.41  | 1.01  | 1.24  | 0.31  |
|              | Err-order | 1.63  | 3.68  | 3.11  | 4.29  |

Table 6: The errors for Triplet on the test set of $D_{20k}$ dataset.

Conclusion

In this work, we propose SentiPrompt to utilize sentiment knowledge enhanced prompts for the ABSA task, which is a task-adaptive modification to a unified generative framework. We inject sentiment knowledge regarding aspects, opinions, and polarities from ground truth into prompts and explicitly model term relations via consistency and polarity judgment templates. Through SentiPrompt MLM during training, the encoder part of the framework learn to explicitly model sentiment relations between aspects and opinions. Likewise, we introduce sentiment knowledge for accurate term extraction and sentiment classification by this tuning process. Additionally, due to mistakes and incompleteness in annotations, we revise four public datasets to form our version $D_{21}$. We conduct experiments on three versions of four benchmark datasets. SentiPrompt achieves significant improvements on ABSA subtasks on all versions of datasets.

\[ We randomly generate 5 numbers in the range of (0, 6000) from the random function. Here $D_{seed} = \{544, 3210, 8, 5678, 744\} \]
References
Bi, Z.; Zhang, N.; Ye, G.; Yu, H.; Chen, X.; and Chen, H. 2021. Intervenional Aspect-Based Sentiment Analysis. arXiv preprint arXiv:2104.11681.

Brown, T.; Mann, B.; Ryder, N.; Subbiah, M.; Kaplan, J. D.; Dhariwal, P.; Neelakantan, A.; Shyam, P.; Sastry, G.; Askell, A.; Agarwal, S.; Herbert-Voss, A.; Krueger, G.; Henighan, T.; Child, R.; Ramesh, A.; Ziegler, D.; Wu, J.; Winter, C.; Hesse, C.; Chen, M.; Sigler, E.; Litwin, M.; Gray, S.; Chess, B.; Clark, J.; Berner, C.; McCandlish, S.; Radford, A.; Sutskever, I.; and Amodei, D. 2020. Language Models are Few-Shot Learners. In Larochelle, H.; Ranzato, M.; Hadsell, R.; Balcan, M. F.; and Lin, H., eds., Advances in Neural Information Processing Systems, volume 33, 1877–1901. Curran Associates, Inc.

Chen, C.; Teng, Z.; and Zhang, Y. 2020. Inducing Target-Specific Latent Structures for Aspect Sentiment Classification. In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), 5596–5607. Online: Association for Computational Linguistics.

Chen, S.; Wang, Y.; Liu, J.; and Wang, Y. 2021a. Bidirectional Machine Reading Comprehension for Aspect Sentiment Triplet Extraction. In AAAI.

Chen, X.; Xie, X.; Zhang, N.; Yan, J.; Deng, S.; Tan, C.; Huang, F.; Si, L.; and Chen, H. 2021b. Adaprompt: Adaptive prompt-based finetuning for relation extraction. arXiv preprint arXiv:2104.07650.

Chen, X.; Zhang, N.; Li, L.; Xie, X.; Deng, S.; Tan, C.; Huang, F.; Si, L.; and Chen, H. 2021c. LightNER: A Lightweight Generative Framework with Prompt-guided Attention for Low-resource NER. arXiv preprint arXiv:2109.00720.

Chen, Z.; and Qian, T. 2020. Relation-Aware Collaborative Learning for Unified Aspect-Based Sentiment Analysis. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, 3685–3694. Online: Association for Computational Linguistics.

Davison, J.; Feldman, J.; and Rush, A. 2019. Commonsense Knowledge Mining from Pretrained Models. Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP).

Ding, N.; Chen, Y.; Han, X.; Xu, G.; Xie, P.; Zheng, H.-T.; Liu, Z.; Li, J.; and Kim, H.-G. 2021. Prompt-Learning for Fine-Grained Entity Typing. arXiv preprint arXiv:2108.10604.

Fan, Z.; Wu, Z.; Dai, X.-Y.; Huang, S.; and Chen, J. 2019. Target-oriented Opinion Words Extraction with Target-fused Neural Sequence Labeling. In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), 2509–2518. Minneapolis, Minnesota: Association for Computational Linguistics.

Gao, T.; Fisch, A.; and Chen, D. 2020. Making Pretrained Language Models Better Few-shot Learners. ArXiv, abs/2012.15723.

Han, X.; Zhao, W.; Ding, N.; Liu, Z.; and Sun, M. 2021. PTR: Prompt Tuning with Rules for Text Classification. CoRR, abs/2105.11259.

He, R.; Lee, W. S.; Ng, H. T.; and Dahlmeier, D. 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, 504–515. Florence, Italy: Association for Computational Linguistics.

Hu, S.; Ding, N.; Wang, H.; Liu, Z.; Li, J.; and Sun, M. 2021. Knowledgeable Prompt-tuning: Incorporating Knowledge into Prompt Verbalizer for Text Classification. arXiv preprint arXiv:2108.02035.

Jiang, Z.; Xu, F. F.; Araki, J.; and Neubig, G. 2020. How Can We Know What Language Models Know? Transactions of the Association for Computational Linguistics, 8: 423–438.

Lewis, M.; Liu, Y.; Goyal, N.; Ghazvininejad, M.; Mohamed, A.; Levy, O.; Stoyanov, V.; and Zettlemoyer, L. 2020. BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension. Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics.

Li, K.; Chen, C.; Quan, X.; Ling, Q.; and Song, Y. 2020. Conditional Augmentation for Aspect Term Extraction via Masked Sequence-to-Sequence Generation. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, 7056–7066. Online: Association for Computational Linguistics.

Li, R.; Chen, H.; Feng, F.; Ma, Z.; Wang, X.; and Hovy, E. 2021. Dual Graph Convolutional Networks 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), 6319–6329. Online: Association for Computational Linguistics.

Li, X.; Bing, L.; Li, P.; Lam, W.; and Yang, Z. 2018. Aspect Term Extraction with History Attention and Selective Transformation. In Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI-18, 4194–4200. International Joint Conferences on Artificial Intelligence Organization.

Li, X. L.; and Liang, P. 2021. Prefix-Tuning: Optimizing Continuous Prompts for Generation. ArXiv, abs/2101.00190.

Liu, X.; Zheng, Y.; Du, Z.; Ding, M.; Qian, Y.; Yang, Z.; and Tang, J. 2021. GPT Understands, Too. arXiv:2103.10385.

Ma, D.; Li, S.; Wu, F.; Xie, X.; and Wang, H. 2019. Exploring Sequence-to-Sequence Learning in Aspect Term Extraction. In Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, 3538–3547. Florence, Italy: Association for Computational Linguistics.

Mao, Y.; Shen, Y.; Yu, C.; and Cai, L. 2021. A Joint Training Dual-MRC Framework for Aspect Based Sentiment Analysis. In AAAI.

Peng, H.; Xu, L.; Bing, L.; Huang, F.; Lu, W.; and Si, L. 2020. Knowing What, How and Why: A Near Complete
Solution for Aspect-Based Sentiment Analysis. *Proceedings of the AAAI Conference on Artificial Intelligence, 34(05): 8600–8606.*

Petroni, F.; Rocktäschel, T.; Riedel, S.; Lewis, P.; Bakhtin, A.; Wu, Y.; and Miller, A. 2019. Language Models as Knowledge Bases? *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP).*

Pontiki, M.; Galanis, D.; Pavlopoulos, J.; Papageorgiou, H.; Androutsopoulos, I.; and Manandhar, S. 2014. SemEval-2014 Task 4: Aspect Based Sentiment Analysis. In *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014),* 27–35. Dublin, Ireland: Association for Computational Linguistics.

Pouran Ben Veyseh, A.; Nouri, N.; Demoncourt, F.; Dou, D.; and Nguyen, T. H. 2020. Introducing Syntactic Structures into Target Opinion Word Extraction with Deep Learning. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP),* 8947–8956. Online: Association for Computational Linguistics.

Schick, T.; and Schutze, H. 2021. Exploiting Cloze-Questions for Few-Shot Text Classification and Natural Language Inference. In *EACL.*

Shin, T.; Razeghi, Y.; Logan IV, R. L.; Wallace, E.; and Singh, S. 2020. AutoPrompt: Eliciting Knowledge from Language Models with Automatically Generated Prompts. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP).*

Talmor, A.; Elazar, Y.; Goldberg, Y.; and Berant, J. 2020. oLMpics-On What Language Model Pre-training Captures. *Transactions of the Association for Computational Linguistics,* 8: 743–758.

Tang, D.; Qin, B.; Feng, X.; and Liu, T. 2016. Effective LSTM for Target-Dependent Sentiment Classification. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers,* 3298–3307. Osaka, Japan: The COLING 2016 Organizing Committee.

Tang, H.; Ji, D.; Li, C.; and Zhou, Q. 2020. Dependency Graph Enhanced Dual-transformer Structure for Aspect-based Sentiment Classification. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics,* 6578–6588. Online: Association for Computational Linguistics.

Vinyals, O.; Fortunato, M.; and Jaitly, N. 2015. Pointer Networks. In Cortes, C.; Lawrence, N.; Lee, D.; Sugiyama, M.; and Garnett, R., eds., *Advances in Neural Information Processing Systems,* volume 28. Curran Associates, Inc.

Wang, K.; Shen, W.; Yang, Y.; Quan, X.; and Wang, R. 2020. Relational Graph Attention Network for Aspect-based Sentiment Analysis. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics,* 3229–3238. Online: Association for Computational Linguistics.

Wang, W.; and Pan, S. J. 2018. Recursive Neural Structural Correspondence Network for Cross-domain Aspect and Opinion Co-Extraction. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers),* 2171–2181. Melbourne, Australia: Association for Computational Linguistics.

Wang, W.; Pan, S. J.; Dahlmeier, D.; and Xiao, X. 2017. Coupled Multi-Layer Attentions for Co-Extraction of Aspect and Opinion Terms. In *AAAI.*

Wang, Y.; Huang, M.; Zhu, X.; and Zhao, L. 2016. Attention-based LSTM for Aspect-level Sentiment Classification. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing,* 606–615. Austin, Texas: Association for Computational Linguistics.

Wu, Z.; Ying, C.; Zhao, F.; Fan, Z.; Dai, X.; and Xia, R. 2020a. Grid Tagging Scheme for Aspect-oriented Fine-grained Opinion Extraction. In *Findings of the Association for Computational Linguistics, EMNLP 2020,* 2576–2585. Online: Association for Computational Linguistics.

Wu, Z.; Zhao, F.; Dai, X.-Y.; Huang, S.; and Chen, J. 2020b. Latent Opinions Transfer Network for Target-Oriented Opinion Words Extraction. *Proceedings of the AAAI Conference on Artificial Intelligence, 34(05): 9298–9305.*

Xu, L.; Chia, Y. K.; and Bing, L. 2021. Learning Span-Level Interactions for Aspect Sentiment Triplet Extraction. 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),* 4755–4766. Online: Association for Computational Linguistics.

Xu, L.; Li, H.; Lu, W.; and Bing, L. 2020. Position-Aware Tagging for Aspect Sentiment Triplet Extraction. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP),* 2339–2349. Online: Association for Computational Linguistics.

Yan, H.; Dai, J.; Qiu, X.; Zhang, Z.; et al. 2021. A Unified Generative Framework for Aspect-Based Sentiment Analysis. *arXiv preprint arXiv:2106.04300.*

Zhang, N.; Deng, S.; Cheng, X.; Chen, X.; Zhang, Y.; Zhang, W.; Chen, H.; and Center, H. I. 2021a. a. Drop Redundant, Shrink Irrelevant: Selective Knowledge Injection for Language Pretraining. In *IJCAI.*

Zhang, N.; Jia, Q.; Deng, S.; Chen, X.; Ye, H.; Chen, H.; Tou, H.; Huang, G.; Wang, Z.; Hua, N.; et al. 2021b. Al-iCG: Fine-grained and Evolvable Conceptual Graph Construction for Semantic Search at Alibaba. *arXiv preprint arXiv:2106.01686.*

Zhang, N.; Li, L.; Chen, X.; Deng, S.; Bi, Z.; Tan, C.; Huang, F.; and Chen, H. 2021c. Differentiable Prompt Makes Pre-trained Language Models Better Few-shot Learners. *arXiv preprint arXiv:2108.13161.*
A. Hyper-parameter settings in sour experiments

All experiments are completed on a single 32G Nvidia-V100 GPU. It takes about 2 hours to finish 50 epochs. Our model is constructed with BART-base pre-trained model. The hyper-parameters to get the best results on every dataset are as in Table 7.

| Dataset  | 14lap | 14res | 15res | 16res |
|----------|-------|-------|-------|-------|
| Training Epochs | 50    |       |       |       |
| Batch Size | 8     | 32    |       |       |
| Learning Rate | 5e-5  | 1e-4  |       |       |
| Hidden Size | 1024  |       |       |       |

Table 7: Hyper-parameters used to get the best performance in training

For AESC, Pair and Triplet, a prediction result is correct only when all the span boundaries are exactly matched and the predicted sentiment polarity are accurately classified. For $D_{20b}$, We report F1 scores of AESC, Pair and Triplet tasks for all datasets. For $D_{20b}$ and $D_{21}$, we report the precision (P), recall (R), and F1 scores of Triplet task for them.

B. Experimental results under few-shot setting scenario

![Figure 5: Comparison of F1 scores for Triplet on the $D_{20b}$ and $D_{21}$ datasets under few-shot setting with 10% of data.]

C. Experimental results for different prompts

We compare prompts with different $N$ and the handcrafted one. The results are shown in Figure 6.

D. Examples of our revision to the annotations in $D_{20b}$

Due to the three problems in previous datasets mentioned in the paper, we revise some annotations in $D_{20b}$ to form our version $D_{21}$. To get these problems through, we pick some typical annotations in $D_{20b}$ and our corresponding revised version as examples in Table 8.

E. Statistical comparison between three versions of four datasets.

A detailed statistical comparison between $D_{20a}$, $D_{20b}$ and $D_{21}$ is presented in Table 9. $D_{21}$ has a bit more sentences and triplets annotations than $D_{20b}$ due to our revision.

F. Comparison between different baselines in our experiments

To make fair comparison between baselines used in our experiments, we summarize the their task formulations, the backbone in their models, and the subtasks they can do in Table 10.

G. The decoding algorithm for converting process

The decoding algorithm for converting the predicted index sequence to aspect spans, opinion spans and polarities is shown in Algorithm 1.

Algorithm 1: Decoding Algorithm for the Triplet

**Require:** $n$, the number of tokens in $X$; target sequence $Y = [y_1, ..., y_n]$; and we have $y_i \in [1, n+1]$

**Ensure:** Target span set $L = \{(a_{i}^{\text{start}}, a_{i}^{\text{end}}, o_{i}^{\text{start}}, o_{i}^{\text{end}}, s_{i})\}$

1: $L = \{\}$
2: while $i <= m$ do
3:   $y_i = Y[i]$
4:   if $y_i > n$ then
5:     $L.add((e, C_{y_i-n}))$
6:   else
7:     $e.append(y_i)$
8:   end if
9: end while
10: return $L$
Sentence: The battery lasts as advertised (give or take 15-20 minutes), and the entire user experience is very elegant.

We ordered a tuna melt-it came with out cheese which just made it a tuna sandwich.

What can you say about a place where the waitress brings out the wrong entree, then verbally assaults your 80 year old grandmother and gives her lip about sending it back (which she did politely, by the way).

In $D_{20b}$ (user experience, elegant, POS) (tuna melt, with out, NEG) (cheese, with out, NEG) (tuna sandwich, with out, NEG) (entree, wrong, NEU)

In $D_{21}$ (battery, lasts as advertised, POS) (user experience, elegant, POS) (tuna melt, with out cheese, NEG)

Table 8: Some examples of our revision to the annotations of samples. In $D_{20b}$, some aspect-opinion pairs are missing and some are wrongly matched. There also exist some complex cases not properly annotated. While in $D_{21}$, we correct those mistakes and use long phrases such as verb-object phrase to annotate complex samples.

| Dataset | 14res | 14lap | 15res | 16res |
|---------|-------|-------|-------|-------|
|         | $N_s$ | $N_p$ | $N_s$ | $N_p$ | $N_s$ | $N_p$ | $N_s$ | $N_p$ |
| $D_{20a}$ train | 1266 | 906 | 1465 | 842 | 1289 |
| dev | 310 | 219 | 346 | 210 | 339 |
| test | 492 | 328 | 543 | 326 | 514 |
| $D_{20b}$ train | 1266 | 906 | 1465 | 842 | 1289 |
| dev | 310 | 219 | 346 | 210 | 339 |
| test | 492 | 328 | 543 | 326 | 514 |
| $D_{21}$ train | 1266 | 906 | 1465 | 842 | 1289 |
| dev | 310 | 219 | 346 | 210 | 339 |
| test | 492 | 328 | 543 | 326 | 514 |

Table 9: The statistics of four datasets, where the $N_s$, $N_p$ denote the numbers of sentences, aspect terms, opinion terms, and the <$a$, $o$> pairs, respectively. These three versions of four datasets can be applied to all ABSA subtasks mentioned above, including AE, OE, ALSC, AOE, AESC, Pair and Triplet.

| Baselines | E2E | Task Formulation | Backbone | AE | OE | ALSC | AOE | AESC | Pair | Triplet |
|-----------|-----|------------------|----------|----|----|------|-----|------|------|--------|
| RINANTE+  | -   | Seq.Tagging      | LSTM+CRF | ✓  | ✓  | ✓    | ✓   | ✓    | ✓    | ✓      |
| CMLA+     | -   | Seq.Tagging      | Attention| ✓  | ✓  | ✓    | ✓   | ✓    | ✓    | ✓      |
| Li-unified+ | -   | Seq.Tagging      | LSTM    | ✓  | ✓  | ✓    | ✓   | ✓    | ✓    | ✓      |
| Peng-two-stage | -   | Seq.Tagging      | LSTM+GCN| ✓  | ✓  | ✓    | ✓   | ✓    | ✓    | ✓      |
| JET-BERT  | ✓   | Seq.Tagging      | BERT    | ✓  | ✓  | ✓    | ✓   | ✓    | ✓    | ✓      |
| Dual-MRC  | ✓   | Span.MRC         | BERT    | ✓  | ✓  | ✓    | ✓   | ✓    | ✓    | ✓      |
| OTE-MTL   | ✓   | Seq.Tagging      | LSTM    | ✓  | ✓  | ✓    | ✓   | ✓    | ✓    | ✓      |
| GTS-BERT  | ✓   | Grid.Tagging     | BERT    | ✓  | ✓  | ✓    | ✓   | ✓    | ✓    | ✓      |
| JET-BERT  | ✓   | Seq.Tagging      | BERT    | ✓  | ✓  | ✓    | ✓   | ✓    | ✓    | ✓      |
| SPAN-BART | ✓   | Span.Generation  | BART    | ✓  | ✓  | ✓    | ✓   | ✓    | ✓    | ✓      |
| Ours      | ✓   | Span.Generation  | BART    | ✓  | ✓  | ✓    | ✓   | ✓    | ✓    | ✓      |

Table 10: The baselines in our experiments. “E2E” is short for End-to-End, which means the model should output all the subtasks’ results synchronously rather than requiring any preconditions, e.g., pipeline methods.