PromptASTE: Prompting a Dataset from Pre-trained Language Models for Unsupervised Aspect Sentiment Triplet Extraction

Anonymous ACL submission

Abstract

Aspect sentiment triplet extraction (ASTE) is a sentiment analysis task that aims to extract views' sentiment polarity, expression, and target (aspect). While the zero-shot scenario for the sentence or aspect-level sentiment has made much progress in recent years, zero-shot ASTE remains unstudied because of its far more complex data structure. This paper challenges this remaining problem and proposes the first unsupervised method for aspect sentiment triplet extraction, which even does not require any training on human-annotated data. Based on the previous discovery of the pre-trained language model (PLM)'s awareness of sentiment, we further leverage the masked language model (MLM) to prompt an ASTE dataset with automatically annotated labels. Our method, PromptASTE, fills in a series of prompts to generate a dataset for related aspects and views. The dataset is then used to train an ASTE model for prediction. Training on PromptASTE results in models with an outstanding capability in discerning sentiment polarities and targeted aspects. Our model sets the first and strong baseline on unsupervised ASTE.

1 Introduction

Aspect sentiment triplet extraction (ASTE) is a type of sentiment analysis task. While conventional sentiment analysis either classifies the sentiment polarity of a sentence or extracts aspect span with polarity, ASTE is interested in aspect-based sentiment and extracts the expression (view) and target (aspect) of sentiments, making it a challenging problem with the complex data structure.

Some instances of ASTE are shown in Figure 1, the view and aspect are represented by spans. Paired spans are labeled as the sentiment polarity of the view on its targeted aspect. While many previous works have been done for the supervised ASTE system, unsupervised ASTE remains a blank. Also, some tries have been made for zero-shot sentence-level and aspect-level sentiment analysis (Sarkar et al., 2019; Wang and Ji, 2022; Phan et al., 2021), but the rather complex data structure of ASTE block these methods from stepping further. As sentiment is a universal and cross-language phenomenon, unsupervised ASTE is appealing to reduce the burden for annotation, especially for low-resource language with a limited number of skilled annotators.

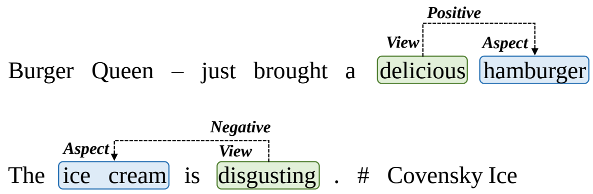

Figure 1: Instances for the ASTE task.

However, unsupervised ASTE is challenging as ASTE data are structured in a complex form. The unsupervised system faces several essential problems for relationship understanding. **a) Polarity** How does the model understand the sentiment polarity with no annotated knowledge? **b) Relationship** How does the model learns paired feature that does not exist in sequential natural language with no annotation for relationships? **c) Boundary** How does the model determine the span boundaries annotated by a human when testing?

The challenges above hinder the application of conventional unsupervised methods, like clustering. Moreover, clustering requires collecting unannotated data for unsupervised training, which is still unfriendly for low-resource languages. We aim to step even further towards a method that is free from any ASTE-related data, no matter annotated or unannotated.

Thus, we cast our attention to pre-train language models (PLMs) (Radford et al., 2018; Devlin et al., 2019; Liu et al., 2019; Yang et al., 2019), which are competitive zero-shot learners (Radford et al.,

2018) with strong scalability. PLMs, like RoBERTa (Liu et al., 2019), are trained on upstream masked language model (MLM) tasks that require the language model to fill in masked words in context. Recent studies have shown that pre-training endows PLMs with sentiment awareness to solve conventional sentiment analysis problems, suggesting the PLM is an admirable choice for unsupervised ASTE. By utilizing the MLM task, we fill in prompts to create an ASTE dataset from PLMs. A prompt combination is used to sample **kernel spans**, which are spans consisting of aspect sentiment triplets, from PLMs.

The annotating system comprises three prompts for domain specification, aspect generation, and view generation. We also propose a contrastive prompt to prompt better sentiment expressions by contrasting positive and negative expressions. Based on the kernel span, PLMs are again used to supplement the contextual background via mask filling. The supplemented data finally form the PromptASTE dataset.

After the dataset is created, PromptASTE is used to train ASTE models following a supervised scenario. Spans and their relationships are annotated in graphs to train an extractor for graphic pattern capturing. We test the trained extractor on several ASTE datasets and compare the results with supervised results. Our method shows competitive performance on unsupervised ASTE and sets the first and strong baseline.

The contributions from our work are summarized as follows:

- We propose the first unsupervised method for ASTE and set a strong baseline for the task.
- We verify the plausibility of prompting a dataset for a task with a complex data structure.
- We implement a novel contrastive prompting procedure to generate sentiment expressions better.

2 Background and Related Work

Triplets in ASTE are formalized in (V, A, P) where V, A, P refer to view (expression) span, aspect (target) span, and sentiment polarity respectively. ASTE models are trained to determine the boundary of spans and label the polarity held by the view towards the aspect.

Since the annotation of a variety of ASTE datasets (Peng et al., 2020; Xu et al., 2020) based on aspect based sentiment analysis (ABSA) data (Pontiki et al., 2014, 2015, 2016), many supervised methods have been proposed for ASTE. (Peng et al., 2020) tests a wide range of previous triplet extracting method on ASTE and propose a tag-and-pair pipeline to set the first supervised baseline. Spans are extracted by finding segments and their representations are fed into a pair classifier to find whether a relationship exists between them. (Xu et al., 2020) incorporates position information and CRF inference into the tagging system to boost performance. (Wu et al., 2020) formalizes ASTE in a grid tagging scheme. The tagged grid is decoded by first finding terms in the diagnosis and then searching for grids indicating relationships between terms. Though supervised ASTE has been under heated discussion since the task's proposal, so far no attention has been cast to solve ASTE with no supervision.

However, unsupervised ASTE is a fairly challenging task. Besides its complex structured nature, the difficulty also comes from the incapability of existing unsupervised systems to build a complete pipeline, from span extraction to relationship labeling. For unsupervised relation extraction, current models have only limited capability to label the relationships between paired already extracted spans (Tran et al., 2020; Yuan and Eldardiry, 2021). These methods use the conventional unsupervised method like clustering to assign closely distributed span pairs to the same labels. Thus, the prerequisite of annotated spans makes these unsupervised methods unfriendly to real zero-shot learning.

Thus, we abandon the conventional unsupervised methods and turn towards leveraging PLMs, which are powerful zero-shot learners via training on super-large corpora. The long training procedure endows PLMs with the understanding of semantic relationships between tokens, which makes the PLM a desirable tool for unsupervised downstream tasks. Also, mask filling on prompts has been verified to be a powerful way to extract commonsense knowledge (Petroni et al., 2019), relationship understanding (Goswami et al., 2020), and sentiment awareness (Wu et al., 2019) of the PLM. Our work further leverages the endowed sentiment awareness in PLMs to build a complete unsupervised pipeline for ASTE.

3 Prompting ASTE Dataset

3.1 The Pipeline

We first give a rough description of our method and how it deals with the challenges in unsupervised ASTE before introducing the specific implementation. The pipeline comprises two main procedures: kernel span generation and context supplement.

Kernel span refers to the span that consists of the aspect sentiment triplet. To obtain kernel spans, our prompt involves masked view spans (v-mask) and masked aspect spans (a-mask). V-masks and a-masks are both common mask tokens used in the upstream MLM pre-training, and their only difference is representing views or aspects. The PLM fills the masked spans, and the kernel span is seized from the span for context supplement.

Polarity We add hints for polarity to the prompt in order to generate view expressions with the corresponding sentiment polarity.

Relationship The relationships are pre-defined between views and aspects in the prompt.

Boundary Words near the span boundaries help control the generated span to have boundaries as pre-defined in the prompt.

Based on the kernel spans, we again use the PLM to supplement the contextual background for the sentiment via mask filling. The supplemented results are the final PromptASTE dataset.

3.2 Domain Prefix Prompt

The domain prefix prompt is used to specify the domain for kernel span generation. As in the green frame in Figure 2, the domain prefix prompt determines the contextual environment for the prompting generation. As the testing datasets are in different domains, the domain prefix prompt will help generate more relevant training data to improve the performance of trained models.

3.3 Aspect Prompt

The aspect prompt is the blue frame in Figure 2, which is responsible for polarity selection and aspect generation. The prompt contains a-masks and a polarity token <*pol*> that provides hints for the later generation.

After the polarity of triplets in the kernel span is selected, the polarity token is substituted by a token with sentiment information. In the instances in Figure 2, the word *good* substitutes <*pos*> and indicates the positive sentiment in the kernel span.

Then we fill in the a-masks via a beam search. Notice that the masked aspect span might consist of multiple mask tokens.

$$
\begin{aligned}
&X = [x_{1:i-1}, \textit{<mask>}, \cdots, \textit{<mask>}, x_{j+1:n}] \\
&p(x_{i:j}|X) = \prod_{t=i}^{j} p(x_t|X, x_{i:t-1}) \\
&p(x_t|X, x_{i:t-1}) = \text{softmax}(R_t/T) \\
&R = \text{PLM}(x_t|X, x_{i:t-1})
\end{aligned}
$$

where X is a sentence with n words and $X_{i:j}$ denotes the span from the i-th word to the j-th word. T refers to the temperature for sampling. $R \in \mathbb{R}^{n\times o}$ is the output representation from the PLM, and o refers to the dictionary size. We summarize the beam searching procedure as Beam$(\cdot)$. After we get the existing probability of each beam, we sample an aspect span following the predicted distribution.

3.4 Contrastive View Prompt

After generating the aspect span, we also fill in the coreference masked aspect span in the view prompt. Then we introduced our contrastive generation for view span.

For the prompt in this step X^{self}, we shift the word in the position of the polarity token to create an opposite prompt X^{oppo}. We first use X^{self} to sample k view span beams by prompting and then calculate the probability distribution of the view span in X^{oppo}.

$$P^{self} = \text{Beam}(X^{self}), P^{oppo} = \text{Beam}(X^{oppo})$$

Finally, the log probability of P^{self} is subtracted by the weighted log probability of P^{self} and passed through a softmax function for the contrastive distribution.

$$P^{contrast} = \text{softmax}(\log(P^{self}) - w\log(P^{oppo}))$$

Here w is a factor that controls the degree of contrast during the generation. The view span is likely sampled following the predicted distribution as the aspect span.

After aspect and view spans are completely filled, we seize the kernel span and build the triplets using pre-defined relationships.

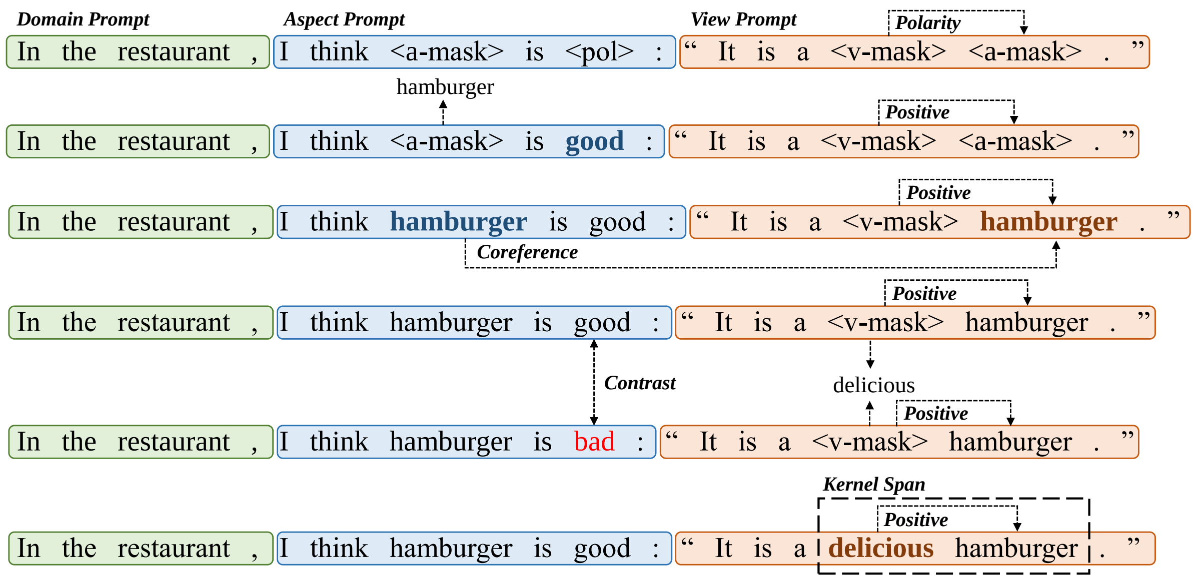

Figure 2: Prompting steps for the generation of PromptASTE.

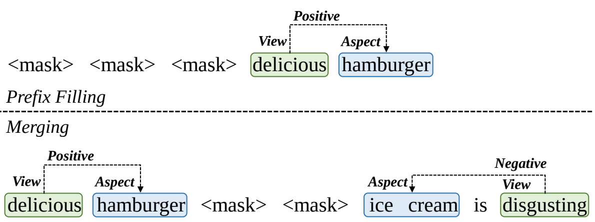

Figure 3: Supplement procedures that transform kernels into training data.

Kernel	Example
<v-mask> <a-mask>	*satisfying service*
<a-mask> is <v-mask>	*screen is fuzzy*
<a-mask> is <v-mask> and <v-mask>	*atmosphere is warm and welcoming*
<a-mask> and <a-mask> are <v-mask>	*smell and taste are good*
<v-mask> <a-mask> and <v-mask> <a-mask>	*nice product and helpful staff*
<v-mask> the <a-mask>	*love the rose*

Figure 4: Kernel spans used in our experiments.

3.5 Context Supplement

Based on the collected kernel spans, we supplement the contextual background for them by continuing to utilize mask filling. We use two supplement scenarios in our experiments: prefix filling and kernel merging as in Figure 3.

Prefix filling is to attach several mask tokens to the beginning of the sentence. Then the PLM fills in the masks following a greedy strategy.

Kernel merging is to merge multiple kernel spans together. We insert several mask tokens between two collected kernels and use the PLM to fill in the mask, still following the greedy strategy.

We avoid adding mask tokens after the kernel span since the generated contents are more likely to break the aspect boundary and generate data with low quality. As a result, we do not apply suffix filling for the context supplement.

4 Experiment

4.1 Testing Data and Metric

We use the ASTE datasets annotated in (Xu et al., 2020) for testing. The datasets include three restaurant review datasets and a laptop review dataset. To compare with previous supervised methods, we use the test datasets for evaluation. Besides, we also create a subset without boundary determination and neutral views to test the model's understanding of relationship and polarity. We first drop all triplets with neutral sentiment polarity. Then, we remove triplets that consist of spans with more than one gram.

For evaluation, we use the F1 score that considers the exact matching of triplets as applied to previous supervised ASTE models. A triplet matches the golden triplet only when their views, aspects, and sentiment polarities are all matched.

Method	14res			14lap			15res			16res		
	P.	R.	F1	P.	R.	F1	P.	R.	F1	P.	R.	F1
(supervised)												
CMLA+	39.18	47.13	42.79	30.09	36.92	33.16	34.56	39.84	37.01	41.34	42.10	41.72
RINANTE+	31.42	39.38	34.95	21.71	18.66	20.07	29.88	30.06	29.97	25.68	22.30	23.87
Li-unified-R	41.04	67.35	51.00	40.56	44.28	42.34	44.72	51.39	47.82	37.33	54.51	44.31
(Peng et al., 2020)	43.24	63.66	51.46	37.38	50.38	42.87	48.07	57.51	52.32	46.96	64.24	54.21
OTE-MTL	63.07	58.25	60.56	54.26	41.07	46.75	60.88	42.68	50.18	65.65	54.28	59.42
JET^{t}	63.44	54.12	58.41	53.53	43.28	47.86	68.20	42.89	52.66	65.28	51.95	57.85
JET^{o}	70.56	55.94	62.40	55.39	47.33	51.04	64.45	51.96	57.53	70.42	58.37	63.83
GTS	71.76	59.09	64.81	57.12	53.42	55.21	54.71	55.05	54.88	65.89	66.27	66.08
(Huang et al., 2021)	63.59	73.44	68.16	57.84	59.33	58.58	54.53	63.30	58.59	63.57	71.98	67.52
(Jing et al., 2021)	67.95	71.23	69.55	62.12	56.38	58.55	60.00	59.27	59.11	70.65	70.23	70.44
(unsupervised)												
MVNA-CT	26.96	32.64	29.53	17.68	22.02	19.61	24.54	27.67	26.01	24.71	30.60	27.34
MVNA-TAG	34.41	41.66	37.69	19.71	24.65	21.90	28.04	30.56	29.25	35.21	42.19	38.29
PromptASTE (res)	**63.80**	35.81	**45.88**	38.71	15.53	22.16	**55.05**	41.15	**47.09**	**60.06**	41.25	**48.90**
PromptASTE (lap)	53.48	35.51	42.68	**40.65**	27.73	**32.97**	46.47	40.34	43.19	56.41	36.72	44.49
PromptASTE (res+lap)	44.69	**42.76**	43.70	36.70	**29.57**	32.75	40.77	**43.71**	42.19	50.16	**46.68**	48.36

Table 1: Main results from our experiments on PromptASTE

4.2 Dataset Configuration

To build the PromptASTE dataset, we design six kernel spans as shown in Figure 4. The whole prompts for kernel construction are shown in Appendix A. Considering the domain variation in the testing dataset, we create two PromptASTE datasets with two different domain prefix prompts as follows.

In the restaurant, ...
For the laptop, ...

The contrastive prompting for a neutral view span is a little different from a positive and negative view. The neutral sentiment does not have a semantically opposite sentiment. Thus, we set both the positive and negative sentiments as the opposite to eliminate the view's polarity. The formula for contrastive generation is rewritten for the neutral view as follows.

$$P^{contrast} = \text{softmax}(\log(P^{self}) - \frac{w}{2}\log(P^{pos})) - \frac{w}{2}\log(P^{neg}))$$

For the generation, we use *RoBERTa-large* as the PLM. Compared to BERT, RoBERTa is pretrained only with the MLM objective, which suggests RoBERTa is able to fully show the potential of a mask-filling-based generation. The beam size is set to 256 to cover a wide range of candidates. Tokens *good*, *bad*, and *average* are used to substitute the polarity token to indicate positive, negative and neutral sentiment polarities. We set temperature T to 1.0 for aspect span generation and 2.5 for context supplement. The temperature for view span generation varies from kernel to kernel to balance the generation's diversity and correctness. The specific setup for these temperatures is included in Appendix B. The weight w for contrastive prompting is 0.6. The max length of the mask token series for context supplement is 6.

4.3 Model and Baseline

Model We take the current state-of-the-art, (Jing et al., 2021) as the learner on our prompt-annotated dataset. (Jing et al., 2021) borrows a combination between table encoder and sequential encoder with interaction from (Wang and Lu, 2020) to build a strong extractor for aspect-view relationships. We completely follow the configuration in the paper to make a direct comparison between models trained on human-annotated and prompt-annotated datasets. We train the model on datasets in the restaurant domain (res), laptop domain (lap), and a combination of two domains (res+lap).

Baseline Because of the lack of unsupervised methods for comparison, we build a simple baseline, matched view, and nearest aspect (MVNA). We use a sentiment dictionary containing positive and negative words from NLTK to match spans in sentiments. The matched spans are taken as view spans with corresponding labels and their nearest noun phrase are extracted as their aspects. We implement two ways to get the noun phrases, using

Method	14res			14lap			15res			16res		
	P.	R.	F1	P.	R.	F1	P.	R.	F1	P.	R.	F1
Supervised	85.97	79.85	82.80	73.18	72.25	72.72	77.62	72.32	74.88	82.08	79.15	80.59
MVNA-CT	38.96	47.10	42.65	22.27	30.63	25.79	33.33	40.11	36.41	34.18	44.13	38.52
MVNA-TAG	54.79	58.71	56.68	34.55	40.86	37.44	43.56	46.01	44.75	51.64	57.49	54.41
PromptASTE (res)	76.06	53.37	62.72	54.76	46.97	50.57	67.74	54.91	60.66	69.37	67.12	68.23
PromptASTE (lap)	61.39	52.27	56.47	52.94	45.25	48.80	60.03	48.17	53.45	64.51	57.85	61.00
PromptASTE (res+lap)	75.81	47.33	58.27	62.64	40.99	49.55	74.19	48.89	58.94	74.19	56.47	64.13

Table 2: Experiment results on the testing data in sampled subsets.

constituency tree (MVNA-CT) or part-of-speech tagger (MVNA) [1]. For MVNA-CT, we sample all noun phrases with no subtree and delete the stop words on each side of the span. For MVNA-TAG, we just sample all continuous *NOUN*-tagged words. To follow up with previous works, we also report the performance of supervised methods to show the remaining gap for zero-shot methods to reach supervised performance.

4.4 Experiment Result

The results from our experiments are presented in Tables 1 and 2. We report the highest results in the experiment. As no unsupervised baseline has been built before, we retrieve results from supervised baselines to evaluate our method's effectiveness.

Main result As in Table 1, we train and test extractor on PromptASTE datasets constructed in different domains. In comparison to unsupervised methods, PromptASTE outperforms the best MVNA generally by 10 F1 scores, verifying its effectiveness as an unsupervised method. PromptASTE achieves precision comparable to recent supervised methods, while recall is the weakness of PromptASTE. This weakness results from the trade-off between generality and simplicity and can be overcome by involving more patterns during prompting. But we want to propose a more general paradigm to prompt unsupervised datasets. Though there still exists a gap between PromptASTE and the highest supervised baseline, the outstanding performance establishes our method as a strong unsupervised baseline.

Domain analysis The main results also show how domain specification in dataset prompting affects the training result. In terms of the F1 score, the extractor performs better when they are trained on prompted data in the same domain as the test data, which is consistent with the research empiric. According to the comparison between extractors trained on datasets with a different domain, and prefix prompts, extractors perform better on in-domain test datasets. Training on data in another generally leads to a drop in both precision and recall, which reflects the penalty from domain difference. The mixture of data from the different domains can improve the recall in the sacrifice of precision by providing various data, which are out-of-domain.

Subset result Table 2 presents the results tested on the sampled datasets. PromptASTE achieves much higher results on the subset due to the difficulty of the unsupervised method to determine boundaries annotated by humans. Free from boundary determination, the gap between PromptASTE and the supervised method is narrowed down in the subset, which better reflects the potential of PLMs for sentiment understanding.

5 Further Analysis

5.1 Few-shot Version

The zero-shot performance of PromptASTE convinces it to be a reasonable method to understand no (annotated) resource circumstance. Here we also consider a less constrained circumstance that we can use a few annotated data as the prompt template for Prompt. We conduct experiments on the 14res dataset by sampling 50 instances.

We set two series of baselines. One is to directly train an extractor based on the few annotated data. The other is to use mask filling (MF) (Kumar et al., 2020) for data augmentation, which is a more straightforward prompting method than PromptASTE. MF$_{view}$ and MF$_{aspect}$ mask-and-fill only the view or aspect span. MF$_{span}$ mask-and-fill both spans and $+aug$ means sampling other 20% words for extra mask-and-filling. When we mask view spans, we attach the label (*positive*, *negative*) of the triplet to the beginning of the sentence with a

[1]We use the tagger and extractor provided by NLTK to get the lexical information.

Dataset	P.	R.	F1	N_{inst}	1-gram(↑)	3-gram(↑)	$SBLEU_2$(↓)	$SBLEU_4$(↓)
14res	67.95	71.23	69.55	2071	14.08	64.20	5.74	2.88
prompted res	66.93	55.21	60.51	7570	19.56	82.30	3.85	1.85
14lap	62.12	56.38	58.55	1456	11.95	56.66	5.58	2.62
prompted lap	65.72	45.22	53.58	3234	17.42	77.90	4.01	1.91

Table 3: Semantic fidelity and diversity of generated data.

Method	P.	R.	F1
(Jing et al., 2021)	48.04	52.99	49.98
MF_{view}	52.32	57.35	54.72
MF_{aspect}	58.17	57.11	57.64
MF_{span}	48.91	63.39	56.88
$MF_{view+aug}$	55.99	56.74	56.36
$MF_{aspect+aug}$	54.72	65.87	59.78
$MF_{span+aug}$	56.23	59.88	58.00
$PromptASTE_z$	63.80	35.81	45.88
$PromptASTE_f$	**69.05**	59.88	64.14
$PromptASTE_{f+z}$	67.30	**64.13**	**65.68**

Table 4: Performance of few-shot PromptASTE.

[SEP] token. We sample 16 times for each instance and apply *RoBERTa-large* for mask filling towards a fair comparison.

Table 4 presents the performance of different few-shot methods. z, f refer to zero-shot and few-shot The state-of-the-art supervised method drops about 20 F1 scores on the few-shot condition, nearly to our zero-shot results. Among the MF methods, mask-and-filling only the aspect span outperforms other methods, indicating generating view span with sentiment polarity. With extra mask-and-filling, the few-shot performance can be further improved as proposed by (Kumar et al., 2020). PromptASTE significantly outperforms the best MF by 4.36 F1 score, verifying its capacity to generate data with better quality. The combination between few-shot and zero-shot PromptASTE further boosts the performance to very close to the supervised performance, showing the potential of PromptASTE in generating human-like annotation.

5.2 Generation Quality

Towards a more comprehensive analysis of our PromptASTE, we also evaluate the quality of instances generated from PromptASTE as we use a generate-and-train strategy. We borrow the evaluating process in (Kumar et al., 2020) for data augmentation, which includes two stages: semantic integrity and diversity.

For semantic integrity, we follow (Kumar et al., 2020) to train an extractor based on the original training dataset and test it on our prompted dataset. We report precision, recall, and F1 score instead of accuracy scores considering the task difference. For diversity, we use the ratio of distinct n-gram (denoted as n-gram) while also including the self BLEU (SBLEU) (Tevet and Berant, 2021) score to provide a broader analysis. The ratio of distinct n-gram is literally the number of distinct n-gram spans divided by the total number of n-gram spans in the dataset. For SBLEU, we sample 1000 sentences from the dataset twice, pair them and then calculate the BLEU scores of the paired sentences. We avoid pairing a sentence to itself and report the average BLEU scores of sentence pairs. For semantic fidelity, we take the results on the test dataset for comparison. For diversity, we use the whole dataset for comparison. The results from our analyses are presented in Table 3.

Semantic Integrity On the prompted dataset, the trained extractor shows a close performance to the original test dataset in precision, while the recall drops by from 10 to 15. The close precision reflects PromptASTE generating data in reliable quality but the relatively low recall discloses the still existing domain difference between the annotated and prompted data. This domain difference also explains why the extractor trained on the prompted dataset achieves lower recall than precision.

Diversity The comparison on diversity shows our prompted data enjoys a higher ratio of distinct n-gram and a lower SBLEU than the human-annotated dataset, indicating the prompted dataset has better diversity in word usage. Thus, the wider coverage of vocabulary is an underlying factor that supports the strong performance of PromptASTE. The reason behind this counter-intuitive

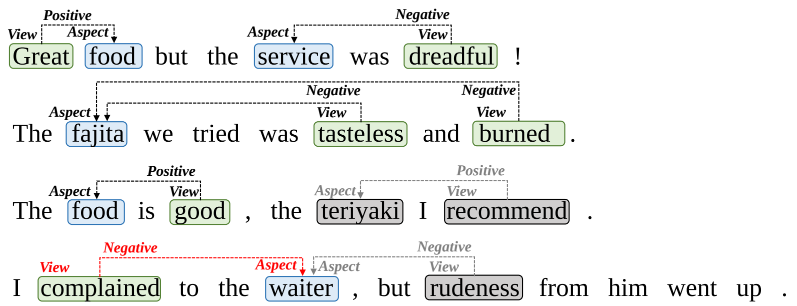

Figure 5: Case Study for the capability boundary of PromptASTE. Grey arrow: Missing triplet (negative false). Red arrow: Incorrect triplet (negative true).

Method	P.	R.	F1
PromptASTE	**76.06**	**53.37**	**62.72**
w/o Domain Prefix	57.65	47.10	51.85
w/o Contrastive Prompting	61.05	53.16	56.83
w/ Suffix Filling	71.21	51.31	59.64

Table 5: Ablation Study on PromptASTE. The subset of res14 is selected as the test dataset.

phenomenon is pre-trained language model learns about various expressions during its training on large-scale corpora while the annotated data only covers a small subset of them. Still, the prompted dataset lacks aspect-view relationship expressions due to constant kernel span forms, but in terms of the lexical level, we conclude prompted data to be more diversified than human-annotated data.

5.3 Ablation Study

To better understand the effects of different modules in our PromptASTE pipeline, we launch an ablation study on them. From the results in Table 5, we can see that domain prefixes and contrastive prompting contribute a lot to the PromptASTE pipeline. Furthermore, We test a pipeline with suffix filling, which fills in mask tokens attached after the kernel span. The performance drop in the ablation study suggests suffix filling is not a beneficial context supplement method. Based on the distribution of kernel spans, the backfire is probably caused by the rather low chance for kernel spans to exist at the beginning of the sentence.

5.4 Case Study

We enumerate and analyze several cases in Figure 5 to specifically show the strength and limitations of PromptASTE.

In the first case, the instance pattern is covered by our prompting pipeline. The instance can be generated by the prompt via kernel merging between two defined kernel spans. As a result, the instance is easily solved by the extractor trained with PromptASTE. The second case shows the scalability of PromptASTE as the pattern of the instance is not covered by prompting. The extractor stays robust against the noise from the adjective component *we tried*. Thus, the triplets are successfully extracted from the sentence. The limitation of PromptASTE is presented in the third case. While the extractor correctly extracts the first triplet, the *recommend-teriyaki* relationship is ignored. As the relationship is in a casual pattern that is very different from our pre-defined ones, the extractor fails to capture it. Incorporating this casual pattern into kernel spans might well solve the problem. The last case includes inference based on coreference, a thorny problem for our parse trained on data with fixed patterns. The case also shows our method to suffer from shortcut learning (Geirhos et al., 2020). The word *complained* is directly recognized as a negative view of the word *waiter*, without understanding the semantic relationships between them. Solving these problems might require pre-trained models for a stronger inference capability.

From the cases, we conclude that our method has some basic understanding of ASTE and enjoys some scalability from the PLM. However, hyperlinguistic phenomena like coreference still remain a problem for us to solve in future studies.

6 Conclusion

We propose a novel method, PromptASTE, for ASTE, which is also the first unsupervised method. We utilize the PLM's understanding of sentiment and apply a series of prompts to construct a training dataset from the PLM. Various prompting mechanisms guarantee the quality of the generated dataset and trained extractor to set a strong baseline for unsupervised ASTE.

References

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 4171–4186. Association for Computational Linguistics.

Robert Geirhos, Jörn-Henrik Jacobsen, Claudio Michaelis, Richard S. Zemel, Wieland Brendel, Matthias Bethge, and Felix A. Wichmann. 2020. Shortcut learning in deep neural networks. *Nat. Mach. Intell.*, 2(11):665–673.

Ankur Goswami, Akshata Bhat, Hadar Ohana, and Theodoros Rekatsinas. 2020. Unsupervised relation extraction from language models using constrained cloze completion. In *Findings of the Association for Computational Linguistics: EMNLP 2020, Online Event, 16-20 November 2020*, volume EMNLP 2020 of *Findings of ACL*, pages 1263–1276. Association for Computational Linguistics.

Lianzhe Huang, Peiyi Wang, Sujian Li, Tianyu Liu, Xiaodong Zhang, Zhicong Cheng, Dawei Yin, and Houfeng Wang. 2021. First target and opinion then polarity: Enhancing target-opinion correlation for aspect sentiment triplet extraction. *CoRR*, abs/2102.08549.

Hongjiang Jing, Zuchao Li, Hai Zhao, and Shu Jiang. 2021. Seeking common but distinguishing difference, A joint aspect-based sentiment analysis model. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 7-11 November, 2021*, pages 3910–3922. Association for Computational Linguistics.

Varun Kumar, Ashutosh Choudhary, and Eunah Cho. 2020. Data augmentation using pre-trained transformer models. *CoRR*, abs/2003.02245.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized BERT pretraining approach. *CoRR*, abs/1907.11692.

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.

Fabio Petroni, Tim Rocktäschel, Sebastian Riedel, Patrick S. H. Lewis, Anton Bakhtin, Yuxiang Wu, and Alexander H. Miller. 2019. Language models as knowledge bases? In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019*, pages 2463–2473. Association for Computational Linguistics.

Khoa Thi-Kim Phan, Duong Ngoc Hao, Dang Van Thin, and Ngan Luu-Thuy Nguyen. 2021. Exploring zero-shot cross-lingual aspect-based sentiment analysis using pre-trained multilingual language models. In *International Conference on Multimedia Analysis and Pattern Recognition, MAPR 2021, Hanoi, Vietnam, October 15-16, 2021*, pages 1–6. IEEE.

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 V. Loukachevitch, Evgeniy V. Kotelnikov, Núria Bel, Salud María Jiménez Zafra, and Gülsen Eryigit. 2016. Semeval-2016 task 5: Aspect based sentiment analysis. In *Proceedings of the 10th International Workshop on Semantic Evaluation, SemEval@NAACL-HLT 2016, San Diego, CA, USA, June 16-17, 2016*, pages 19–30. The Association for Computer 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@NAACL-HLT 2015, Denver, Colorado, USA, June 4-5, 2015*, pages 486–495. The Association for Computer Linguistics.

Maria Pontiki, Dimitris Galanis, John Pavlopoulos, Haris Papageorgiou, Ion Androutsopoulos, and Suresh Manandhar. 2014. Semeval-2014 task 4: Aspect based sentiment analysis. In *Proceedings of the 8th International Workshop on Semantic Evaluation, SemEval@COLING 2014, Dublin, Ireland, August 23-24, 2014*, pages 27–35. The Association for Computer Linguistics.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2018. Language models are unsupervised multitask learners.

Anindya Sarkar, Sujeeth Reddy, and Raghu Sesha Iyengar. 2019. Zero-shot multilingual sentiment analysis using hierarchical attentive network and BERT. In *NLPIR 2019: The 3rd International Conference on Natural Language Processing and Information Retrieval, Tokushima, Japan, June 28 - 30, 2019*, pages 49–56. ACM.

Guy Tevet and Jonathan Berant. 2021. Evaluating the evaluation of diversity in natural language generation. In *Proceedings of the 16th Conference of the*

European Chapter of the Association for Computational Linguistics: Main Volume, EACL 2021, Online, April 19 - 23, 2021, pages 326–346. Association for Computational Linguistics.

Thy Thy Tran, Phong Le, and Sophia Ananiadou. 2020. Revisiting unsupervised relation extraction. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 7498–7505. Association for Computational Linguistics.

Jue Wang and Wei Lu. 2020. Two are better than one: Joint entity and relation extraction with table-sequence encoders. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*, pages 1706–1721. Association for Computational Linguistics.

Zhenhailong Wang and Heng Ji. 2022. Open vocabulary electroencephalography-to-text decoding and zero-shot sentiment classification. In *Thirty-Sixth AAAI Conference on Artificial Intelligence, AAAI 2022, Thirty-Fourth Conference on Innovative Applications of Artificial Intelligence, IAAI 2022, The Twelveth Symposium on Educational Advances in Artificial Intelligence, EAAI 2022 Virtual Event, February 22 - March 1, 2022*, pages 5350–5358. AAAI Press.

Xing Wu, Tao Zhang, Liangjun Zang, Jizhong Han, and Songlin Hu. 2019. Mask and infill: Applying masked language model for sentiment transfer. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI 2019, Macao, China, August 10-16, 2019*, pages 5271–5277. ijcai.org.

Zhen Wu, Chengcan Ying, Fei Zhao, Zhifang Fan, Xinyu Dai, and Rui Xia. 2020. Grid tagging scheme for aspect-oriented fine-grained opinion extraction. *CoRR*, abs/2010.04640.

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 2020, Online, November 16-20, 2020*, pages 2339–2349. Association for Computational Linguistics.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime G. Carbonell, Ruslan Salakhutdinov, and Quoc V. Le. 2019. Xlnet: Generalized autoregressive pretraining for language understanding. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 5754–5764.

Chenhan Yuan and Hoda Eldardiry. 2021. Unsupervised relation extraction: A variational autoencoder approach. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 7-11 November, 2021*, pages 1929–1938. Association for Computational Linguistics.

A Whole Prompt for Kernel Building

We present the whole prompts used in our experiments in Figure 6. Some special tokens are in the prompts. <*prefix*> refers to the domain prefix prompt. <*det*> refers to the determinative component. <*adv*> refers to the adverb component. <*be*> refers to words with the *be* lemma.

B Prompting Configuration

Kernel	**Temperature**
Polarity <v-mask> <a-mask>	3.00
Polarity <a-mask> is <v-mask>	1.50
Polarity *Polarity* <a-mask> is <v-mask> and <v-mask>	1.50
Polarity <a-mask> and <a-mask> are <v-mask>	1.50
Polarity *Polarity* <v-mask> <a-mask> and <v-mask> <a-mask>	3.00
Polarity <v-mask> the <a-mask>	6.00

Figure 7: The configuration for the temperature to generate view spans.

The temperature configuration for prompting is shown in Figure 7.

C Statistical Properties of Datasets

Prop.	14res	15res	16res	14lap
Sent. Num.	2.1k	1.1k	1.4k	1.5k
Sent. Len.	16.9	15.0	14.9	18.4
Span. Num.	6.8k	3.1k	4.0k	4.1k
Span. Len.	1.3	1.3	1.3	1.4
Rel. Num.	4.0k	1.7k	2.2k	2.4k

Table 6: Statistical properties of the triplet parsing datasets used in our experiments.

The statistical properties of the triplet parsing datasets in our experiments are presented in Table 6.

Coreference *Polarity*

<s> <prefix> , the <a-mask> is <pol> : “ <det> <adv> <v-mask> <a-mask> .” </s>

Kernel Span

Coreference *Polarity*

<s> <prefix> , the <a-mask> is <pol> : “ <det> <a-mask> <be> <adv> <v-mask> .” </s>

Kernel Span

Coreference *Polarity* *Polarity*

<s> <prefix> , the <a-mask> is <pol> : “ <det> <a-mask> <be> <adv> <v-mask> and <v-mask> .” </s>

Kernel Span

Coreference *Coreference* *Polarity*

<s> <prefix> , the <a-mask> and <a-mask> are <pol> : “ <det> <a-mask> and <a-mask> <be> <adv> <v-mask> .” </s>

Kernel Span

Coreference *Coreference* *Polarity* *Polarity*

<s> <prefix> , the <a-mask> and <a-mask> are <pol> : “ <det> <adv> <v-mask> <a-mask> and <v-mask> <a-mask> .” </s>

Kernel Span

Coreference *Polarity*

<s> <prefix> , the <a-mask> is <pol> : “ I <v-mask> <det> <a-mask> .” </s>

Kernel Span

Figure 6: The whole format of prompts used in our experiments.