SynPrompt: Syntax-aware Enhanced Prompt Engineering for Aspect-based Sentiment Analysis

Wen Yin[12], **Cencen Liu**[12], **YI XU**[12]*, **Ahmad Wahla**[2], **Huang Yiting**[3], **Dezhang Zheng**[1]
[1]School of Information and Software Engineering,
University of Electronic Science and Technology of China, Chengdu 610054, China
yinwenok,cencenliu,202122090520@std.uestc.edu.cn
[2]Laboratory Of Intelligent Collaborative Computing, Chengdu 611731, China
xuyi0421@uestc.edu.cn, ahmadrazawahla4@gmail.com
[3]Bashu Middle School, Chongqing 400013, China
1020864824@hotmail.com

Abstract

Although there have been some studies uses prompt learning for the Aspect-based Sentiment Analysis(ABSA) tasks, their methods of prompt-tuning are simple and crude. Compared with vanilla fine-tuning methods, prompt learning intuitively bridges the objective form gap between pre-training and fine-tuning. Concretely, simply constructing prompt related to aspect words fails to fully exploit the potential of Pre-trained Language Models, and conducting more robust and professional prompt engineering for downstream tasks is a challenging problem that needs to be solved urgently. Therefore, in this paper, we propose a novel syntax-aware enhanced prompt method (SynPrompt), which sufficiently mines the key syntactic information related to aspect words from the syntactic dependency tree. Additionally, to effectively harness the domain-specific knowledge embedded within PLMs for the ABSA tasks, we constructed two adaptive prompt frameworks to enhance the perception ability of the above method. After conducting extensive experiments on three benchmark datasets, we have found that our method consistently achieves favorable results. These findings not only demonstrate the effectiveness and rationality of our proposed methods but also provide a powerful alternative to traditional prompt-tuning.

Keywords: ABSA, Syntactic Dependency Tree, Prompt Learning

1. Introduction

Aspect-based Sentiment Analysis (ABSA) is a fine-grained branch of sentiment analysis that aims to recognize the sentimental polarity of the specific aspect in a sentence (Jiang et al., 2011). For example, given the sentence "*The food in this restaurant is very nice, but the service is terrible.*", the goal of the ABSA task is to define the emotional polarity of the "*food*" and "*service*" aspects as positive and negative, respectively.

Many previous works (Song et al., 2019; Xu et al., 2019; Karimi et al., 2021)for the ABSA task focus on fine-tuning based on Pre-trained Language Models(PLMs) such as Bidirectional Encoder Representation from Transformers(BERT) (Devlin et al., 2019). It is difficult to determine whether the knowledge that fine-tuned LMs contain is learned during the pretraining or the fine-tuning process (Shin et al., 2020), which restricts PLMs from reaching their full potential. Thus, to mitigate this problem, there are some works (Li et al., 2021a; Gao et al., 2022; Yang and Zhao, 2022; Yin et al., 2023) that introduce prompts learning to the ABSA task. Figure 1 shows how existing work typically uses prompts for the ABSA tasks. In this paradigm, instead of relying on traditional fine-tuning methods, prompt learning involves reformulating downstream tasks to resemble those encountered during the original LM training by utilizing a textual prompt (Liu et al., 2023). However, these studies use rough prompts, which are difficult to adapt to specific task scenarios. Therefore, more robust and professional, **prompt engineering** to promote comprehensive stimulation of PLMs and effective adaptation of specific task is needed, which involves designing task-specific prompt templates and verbalizers.

In light of this, in this work, we propose a Syntax-aware Enhanced Prompt method (**SynPrompt**), which constructs our prompt by looking for words that are related to aspect words syntactic. Specifically, we consider fully exploiting the syntactic information of the syntactic dependency tree (*i.e.*, Dep.Tree) to address the problem. Typically, the syntactic dependency tree contains the dependency relations of individual words in a sentence, which helps to correctly align aspect terms and their corresponding words expressing sentiments (Liang et al., 2022). Dep.Tree can discriminate different relations among aspects to infer the sentiment relations of different aspects. We present the Dep.Tree of the example "The food is great but the service is dreadful." in Figure 2.

Furthermore, the fine-grained nature of ABSA poses difficulties in effectively mining knowledge from language models for practical application. In

*Corresponding author

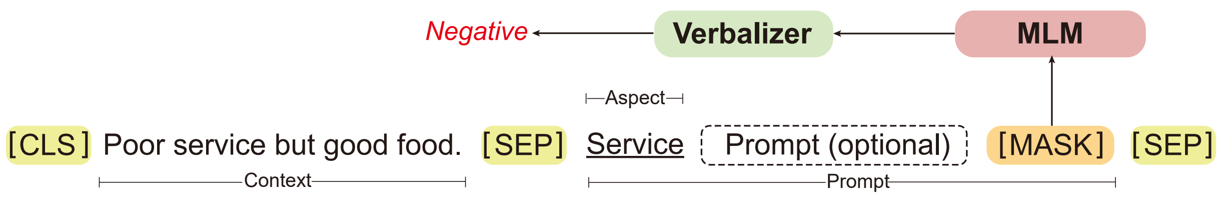

Negative
Verbalizer
MLM
Aspect
[CLS] Poor service but good food. [SEP] Service Prompt (optional) [MASK] [SEP]
Context
Prompt

Figure 1: A general template based on prompt tuning for the ABSA task.

order to improve the robustness and perception of our method, we construct two adaptive prompt frameworks for the ABSA task: the Auto prompt—a discrete prompt framework that can automatically select discrete prompt tokens, and the Soft Prompt—a continuous prompt framework based on learnable pseudo tokens. In general, it is more effective to construct this automatic prompt structure (Shin et al., 2020; Liu et al., 2021), which can further inject and stimulate the task-related knowledge in PLMs, thus boosting the model performance.

Experimentally, we conduct extensive experiments on three widely used datasets in full-shot and few-shot settings. As a result, our proposed SynPrompt method based on two prompt frameworks outperforms the current methods in low-resource scenarios and presents a competitive performance at the same scale in full-data scenarios. Sufficient empirical analysis shows that our proposed methods are a stronger alternative to vanilla fine-tuning or rough prompt-tuning[1].

To sum up, our key contributions are:

- We propose a novel Syntax-aware Enhanced Prompt method based on prompt engineering for the ABSA task, we can design which highlights that more powerful prompt contents can be designed through syntactic relations.
- We built two prompt frameworks for the above methods, which prove the feasibility and rationality of the SynPrompt described above. Our SynPrompt serves as a valuable tool within these frameworks, allowing for enhanced performance and flexibility in ABSA.

2. Related Work

2.1. Aspect-based Sentiment Analysis

Given the large impact that pre-trained LMs have had on NLP in the pre-train and fine-tune paradigm, existing work (Song et al., 2019; Karimi et al., 2021; Li et al., 2021b)for the ABSA task focuses on fine-tuning based on PLMs such as BERT (Devlin et al., 2019). Other work (Liang et al., 2022; Wang et al., 2020; Zhang et al., 2022)involves modeling sentences and aspects based on graph neural networks, which extract critical semantic and syntactic information from constituent trees and dependency trees.

However, existing methods still face challenges in fully harnessing the effective knowledge contained within PLMs due to the discrepancy between pre-training and fine-tuning. To address this limitation, prompt learning has emerged as a promising approach. (Li et al., 2021a) proposes SentiPrompt to use sentiment knowledge enhanced prompts to tune the language model in the unified framework. (Gao et al., 2022) proposes a multi-task framework that can solve multiple ABSA tasks by controlling the type of task prompts consisting of multiple element prompts. (Yin et al., 2023) proposes Prompt-oriented Fine-tuning Dual BERT model that considers the complex semantic relevance and the scarce data samples simultaneously. (Yang and Zhao, 2022) proposes a novel end-to-end framework, where numerous sentiment aspects are elicited by a machine reading comprehension (MRC) model in a prompt learning way.

2.2. Prompt Learning

There has been some work on tips to learn how to improve PLM performance. (Petroni et al., 2019) find that without fine-tuning, BERT contains relational knowledge competitive with traditional NLP methods that have some access to Oracle knowledge. (Gao et al., 2021) presented a suite of simple and complementary techniques for fine-tuning language models on a small number of annotated examples. That approach includes prompt-based fine-tuning together with a novel pipeline for automating prompt generation. (Lester et al., 2021) explored ”prompt tuning,” a simple yet effective mechanism for learning ”soft prompts” to condition frozen language models to perform specific downstream tasks.

Our investigation reveals that, given a suite of appropriate prompts (known as prompt engineering), PLMs can feed back internally specific language knowledge. Recent work has also focused on prompt engineering. (Schick and

[1]Our source code and datasets involved in this paper are released at https://github.com/yinwen2019/Prompt-ABSA

Schütze, 2021) introduced Pattern Exploiting Training (PET), a semi-supervised training procedure that reformulates input examples as cloze-style phrases to help language models understand a given task. (Li and Liang, 2021) propose prefix-tuning, a lightweight alternative to fine-tuning for natural language generation tasks.

However, there has been not enough in-depth research on prompt engineering for the ABSA task. Thus, in this work, we rely on this thinking to make a more adaptive adjustment for the ABSA task.

3. Preliminary

In this section, we first give a problem definition of the ABSA task (§3.1), followed by an introduction to conventional vanilla fine-tuning (§3.2)and prompt-based tuning (§3.3) with PLMs.

3.1. Problem Definition

In this paper, we just focus on Aspect Sentiment Classification (ASC), which is to judge the emotion for a given aspect word in a sentence. The inputs of the ASC task are a sentence and a predefined aspect set ($\mathcal{S}, \mathcal{A}$). We let $\mathcal{S} = \{w_1, w_2, ...w_n\}$ and $\mathcal{A} = \{a_1, a_2, ...a_m\}$ represent a sentence and a predefined aspect set, where n and m are the numbers of words in $\mathcal{S}$ and the number of aspects in $\mathcal{A}$, respectively. For each $\mathcal{S}$, $\mathcal{A}_s = \{a_i | a_i \in \mathcal{A}, a_i \in \mathcal{S}\}$ denotes the aspects contained in $\mathcal{S}$. We treat each multiple-word aspect as a single word for simplicity, so a_i also means the i-th word of $\mathcal{S}$. The goal of ASC is to predict the sentiment polarity $\mathcal{Y} = \{\text{Positive}, \text{Negative}, \text{Neutral}\}$ of the given aspect $a_i \in \mathcal{A}_s$ in the input sentence $x = \{\mathcal{S}, \mathcal{A}_s\}$.

3.2. Vanilla Fine-tuning

In the vanilla fine-tuning paradigm of the ASC task, the input is structured as a concatenation of the original sequence and the aspect word "$x_{\text{ft}} = [\text{CLS}], \mathcal{S}, a_i, [\text{SEP}]$", where $a_i \in \mathcal{A}_s$. In other words, for each aspect word, we construct a sequence as input of the PLM. Empirically, the embedding of the [CLS] token produced by PLM, $\text{h}_{[\text{CLS}]}$, is the classifier token that is fed into an output layer to predict the probability distribution over the label space. The predicted probability distribution is generally calculated by the following formula:

$$P(y|x) = \text{Softmax}(\text{W} \times \text{h}_{[\text{CLS}]} + \text{b}) \qquad (1)$$

where W and b are learnable parameters of PLMs. They are usually fine-tuned by minimizing cross-entropy loss as the objective function.

3.3. Prompt-based Tuning

In particular, prompt-based tuning uses the [MASK] token as a predictor instead of the [CLS] token. It predicts the probability of masked words by using the [MASK] token, which is generally used with the prompt template by prompt engineering. **Prompt engineering** is the process of creating a prompting function $f_{prompt}(x)$ that results in the most effective performance on the downstream task (Liu et al., 2023). Different from vanilla fine-tuning, we get the original input sequence through a prompt engineering $x_{\text{prompt}} = f_{prompt}(x)$, which can transform the original input into the special input using in prompt tuning. In this way, we can use the words predicted by the [MASK] as the input of the downstream task. Moreover, we need to build a verbalizer that maps each label $y \in \mathcal{Y}$ to a set of label vocabularies $\mathcal{V}_y = \{w_1, ...w_k\}$, which are a subset of the vocabulary $\mathcal{V}$ of PLM. For example, we could map a set $\mathcal{V}_y = \{good, fantastic, fine...\}$ into y = Positive. In this case, the class probability distribution is obtained by marginalizing the set of label tokens:

$$P(y|x_{\text{prompt}}) = \sum_{w \in \mathcal{V}_y} p([\text{MASK}] = w | x_{\text{prompt}}) \qquad (2)$$

Of course, the template and the verbalizer are the core of prompt engineering. Therefore, it is meaningful and important to design a framework with appropriate templates and optimizers for downstream tasks.

4. Methodology

In this section, we will introduce how to engineer ABSA tasks using our proposed SynPrompt method (§4.1), then elaborate two novel prompt framework (§4.2) that adapt the above method. Finally, we introduce the verbalizers we used and the training of the model (§4.3)

4.1. Syntax-aware Enhanced Prompt

The overview of the Syntax-aware Enhanced Prompt is shown in Figure 2. We first obtain the corresponding syntactic dependency tree based on the original sentence. Later Syntactic Distance Matrix(§4.1.1) and Syntactic Relation Matrix(§4.1.2) are calculated based on the Dep.Tree and the final prompt decision(§4.1.3) is made based on them. Finally, the selected Prompt word is injected into the Prompt content.

4.1.1. Syntactic Distance Matrix

We treat the syntactic Dep.Tree as a directed graph, and each token as a node. Firstly, we define the distance between node w_i and w_j as

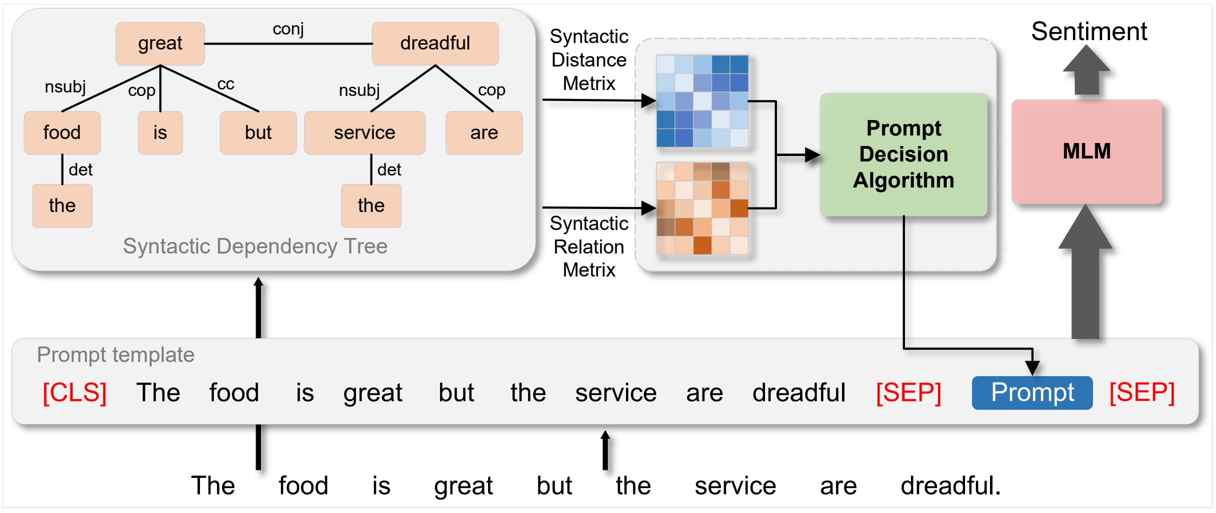
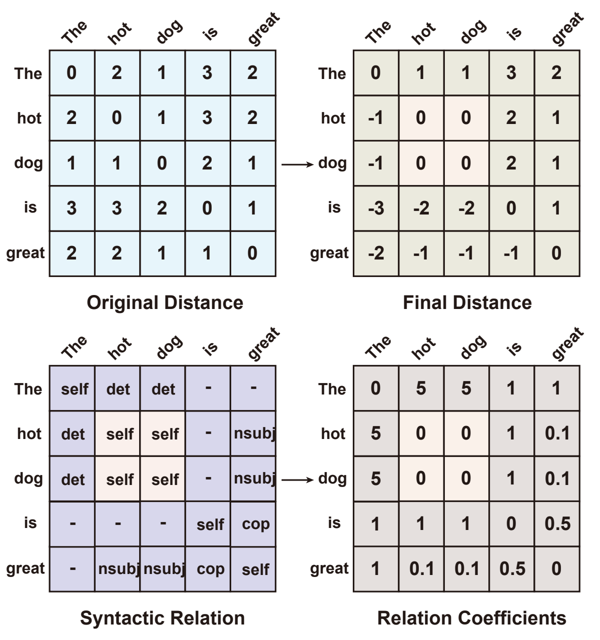

great
conj
dreadful
nsubj
cop
cc
nsubj
cop
food
is
but
service
are
det
det
the
the
Syntactic Dependency Tree
Syntactic Distance Metrix
Syntactic Relation Metrix
Prompt Decision Algorithm
Sentiment
MLM
Prompt template
[CLS] The food is great but the service are dreadful [SEP] Prompt [SEP]
The food is great but the service are dreadful.

Figure 2: The construction process of Syntactic Distance Matrix and Syntactic Relation Matrix targets the example sentence.



Original Distance

	The	hot	dog	is	great
The	0	2	1	3	2
hot	2	0	1	3	2
dog	1	1	0	2	1
is	3	3	2	0	1
great	2	2	1	1	0

Final Distance

	The	hot	dog	is	great
The	0	1	1	3	2
hot	-1	0	0	2	1
dog	-1	0	0	2	1
is	-3	-2	-2	0	1
great	-2	-1	-1	-1	0

Syntactic Relation

	The	hot	dog	is	great
The	self	det	det	-	-
hot	det	self	self	-	nsubj
dog	det	self	self	-	nsubj
is	-	-	-	self	cop
great	-	nsubj	nsubj	cop	self

Relation Coefficients

	The	hot	dog	is	great
The	0	5	5	1	1
hot	5	0	0	1	0.1
dog	5	0	0	1	0.1
is	1	1	1	0	0.5
great	1	0.1	0.1	0.5	0

Figure 3: The construction process of Syntactic Distance Matrix and Syntactic Relation Matrix targets the example sentence.

$d(w_i, w_j)$ that is not positive or negative. We define the original distance matrix as D^*:

$$D^*(i,j) = d(w_i, w_j) \tag{3}$$

Subsequently, determine the positive and negative of these distances according to the order of the words in the original sentence, that is, the distance from the preceding word to thWe following word in the sentence is plus, and the reverse direction is minus. We define the final distance matrix as D:

$$D(i,j) = \begin{cases} D^*(i,j), & i < j \\ -D^*(i,j), & i \geq j \end{cases} \tag{4}$$

Notably, when the aspect word has more than one token, the self-distance is 0, and the positive and negative distances are calculated according to the beginning and end of the aspect word. As shown in the top half of Figure 3, is a construction process of the final distance matrix D targets the sentence "*The hot dog is great*".

4.1.2. Syntactic Relation Matrix

There are many kinds of syntactic dependency relations from Dep.Tree. As a matter of fact, there are some typical relations that determine the polarity of aspect words (e.g., **nsubj** between "*food*" and "*delicious*"). Therefore, we define the relationship coefficient r based on the importance of the syntactic relation for our calculation. Note that a smaller coefficient indicates a more important relation. We obtain the final relation matrix R by mapping each relation to predefined coefficients:

$$R(i,j) = \text{MAP}(Relation(i,j)) \tag{5}$$

where $Relation(i,j)$ represents the syntactic relation of node w_i and node w_j, and MAP stands for the mapping in the appendix. As shown in the bottom half of Figure 3, is a construction process of the relation coefficients matrix D targets the sentence "*The hot dog is great*".

4.1.3. Prompt Decision Algorithm

The details of the prompt decision algorithm are shown in Algorithm 1. Firstly, we compute the result of the dot multiplication of matrix D and R. Note that the result is again the adjacency matrix, which represents the distance of each node on the tree. Immediately following, we compute the shortest distance from each node with respect to

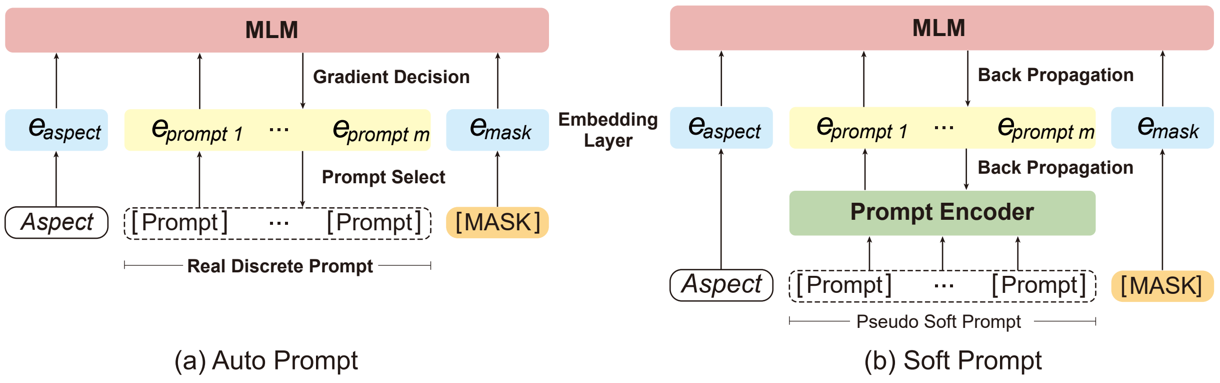


Figure 4: Two automatic prompt frameworks respectively with real discrete prompt and pseudo continuous prompt.

Algorithm 1: Prompt Decision

```
Data: Syntactic distance Matrix $D$ and
      syntactic relation matrix $R$ . The
      index of aspect words $a$ in the
      sentence x . Decision threshold $k$.
Result: Decided prompt word list $\mathcal{P}$ of the
        given matrixes.
Initialize $\mathcal{P} = \{x_a\}, \mathcal{C} = \{\}, t = 0$
Get adjacent matrix $M \leftarrow D \odot R$
Get List of shortest paths about aspect:
 $\mathcal{C} \leftarrow Dijkstra(M, a)$
while $t \leq k$ do
    $i \leftarrow getIndexOfMin(\mathcal{C})$
    if $i < a$ then $\mathcal{P}.insert(\mathrm{x}_i)$
    if $i \geq a$ then $\mathcal{P}.append(\mathrm{x}_i)$
    Update $\mathcal{C}_i \leftarrow$ DBLMAX
    $t \leftarrow t + 1$
return $\mathcal{P}$;
```

the aspect nodes using Dijkstra's algorithm. The word closest to the aspect, which is calculated with threshold k from the obtained distance list $\mathcal{C}$, is determined as the prompt word. These words are added to before and after the aspect words in the prompt word list $\mathcal{P}$ in the order they were in the original sentence. Finally, prompt word list $\mathcal{P}$ is injected into the prompt framework for prompt-tuning.

4.2. Adaptive Prompt Frameworks

In what follows, we present the implementation of two adaptive prompt frameworks for our task: Auto Prompt and Soft Prompt.

4.2.1. Auto Prompt for ABSA

Auto prompt implements an automatic real discrete prompt template design process. Here, as shown in Figure 4 - (a), we propose a method for automatic prompt construction based on (Shin et al., 2020). We begin by adding a number of original [Prompt] tokens to the template between the aspect word and the [MASK] token, which will be replaced over time during training. Note that this token is independent of MLM's vocabulary, which means we need to add the first token. Next, we will introduce a gradient-based prompt automatic selection method.

Gradient-Based Prompt Selection We send the initialized Prompt templates into MLM, which contain the special tokens [Prompt] and [MASK]. Our goal is to find real discrete prompt words that can minimize model losses through gradient calculation. Formally, at each epoch, We calculate the first-order approximate logarithmic likelihood, which is obtained by multiplying the gradient of the embedding layer by the backpropagation of the loss with the word embedding in the vocabulary $\{\boldsymbol{e}_w | w \in \mathcal{V}\}$. Then we identify a candidate set $\mathcal{V}_{cand}$ of the top-k tokens estimated to cause the greatest increase:

$$\mathcal{V}_{cand} = \underset{w \in \mathcal{V}}{\text{top-}k} \left[\boldsymbol{e}_w^\top \nabla \log p\left(y | x_{prompt}\right) \right] \tag{6}$$

where $\boldsymbol{e}_w$ is the input embedding of w, and the gradient is calculated from log-likelihood estimation. In other words, this set of candidates results in the maximum logarithmic likelihood, that is the minimum loss of the model. Subsequently, for each candidate in this set, we re-evaluate Equation (8) based on our framework by replacing the [Prompt] token with the candidate word one by one and retaining the prompt word with the highest probability in the next epoch.

4.2.2. Soft Prompt for ABSA

In contrast to the previously mentioned prompt templates, soft prompts adopt a unique approach by directly prompting the model within its embedding space. Instead of using actual prompt words, soft prompts utilize pseudo-prompt words that exist within the embedded space. These pseudo-prompt words can be adjusted through template parameters, allowing for flexibility and customization. Figure 4 - (b) illustrates the process, where a prompt encoder is employed to automatically learn improved word embedding representations for the prompt words, following the methodology described in (Liu et al., 2021). This technique enables enhanced control and optimization of the prompt information within the embedding space.

Prompt Encoder The Prompt Encoder, a lightweight neural network, plays a crucial role in minimizing the discreteness among prompt words. To achieve this, we initially initialized the [Prompt] tokens randomly and subsequently constructed an encoder comprising a bidirectional Long Short-Term Memory (LSTM) network and a two-layer Multilayer Perceptron (MLP). Formally, we regard the calculation process as:

$$\boldsymbol{e}_i = \text{MLP}(\text{LSTM}\left[\overrightarrow{\boldsymbol{e}_i} : \overleftarrow{\boldsymbol{e}_i}\right]) \tag{7}$$

where $\boldsymbol{e}_i$ represents the initial embedding of the i-th pseudo prompt word. In LSTM, we only need the output embedding e. For the two-layer perceptron, we select the ReLU activation function to optimize the encoder.

4.3. Training

As shown in Figure 1, we take the original sentence as input to the forward sentence. For the backward sentences, we choose aspect words, prompt contents and the [MASK] token as the final prompt sentence. It is important to note that the content of the prompt is optional and configurable for different prompt templates. Subsequently, we select the hidden layer representation of the [MASK] token from the MLM output $\mathbf{h}_{\text{[MASK]}}$, which represents the word probability distribution of the masked word.

Verbalizers Unlike normal Verbalizers, we don't design label vocabulary directly for label classes. We use a multilayer perceptron (MLP) to map the hidden layer representation of the [MASK] token to a low-dimensional vector space. In other words, we map the word probability distribution directly to the class label distribution without having to set a fixed mapping relationship for the label class:

$$\hat{y} = \text{Softmax}(\text{MLP}([\mathbf{h}_{\text{[MASK]}}])) \tag{8}$$

where MLP are learnable parameters and the hidden layer size of input $\mathbf{h}_{\text{[MASK]}}$ is equal to the vocabulary size.

Finally, whether it's an auto prompt or a soft prompt, we apply the cross-entropy loss function for the MLM's parameters training:

$$\mathcal{L} = -\sum_{(s,a)\in\mathcal{D}}\sum_{c\in\mathcal{C}} \log p\left(y|x_{prompt}\right) + \|\theta\|_2^2 \tag{9}$$

Dataset	#positive		#negative		#neutral	
	Train	Test	Train	Test	Train	Test
Restaurant	2164	727	807	196	637	196
Laptop	976	337	851	128	455	167
Twitter	1507	172	1528	169	3016	336

Table 1: Statistics for the three experimental datasets.

Dataset	#shot-type	#1	#2	#4	#8
Restaurant	36	36	72	144	288
Laptop	22	22	44	88	176
Twitter	60	60	120	240	480

Table 2: Statistics on the number of few-hot learning for the three experimental datasets.

where $\mathcal{D}$ contains all the sentence-aspect pairs and a represents the aspect appearing in sentence s. θ represents all the trainable parameters and $\mathcal{C}$ is the collection of sentiment polarities. θ represents all trainable model parameters.

5. Experiments

In order to comprehensively evaluate the effectiveness and rationality of our method, we completed sufficient experimental verification. In the following, we use SynPrompt to denote the syntax-aware enhanced prompt method in §4.1 and use AutoP to denote the Auto Prompt framework in §4.2.1. SoftP denotes the Soft Prompt framework in §4.2.2.

5.1. Datasets

The experiments were conducted on three benchmark ABSA datasets: SemEval 2014 Task 4 Restaurant and Laptop reviews (Pontiki et al., 2014), and Twitter posts (Dong et al., 2014). Each data item was labeled with one of the three sentiment polarities: positive, negative, or neutral. Moreover, we strictly adhere to the dataset configurations of previous studies, utilizing the provided train and test sets without any modifications or additional divisions. The statistical information of three datasets is shown in Table 1.

5.2. Implementation Details

For our experiments, we initialize word embeddings with the official BERT models provided by (Devlin et al., 2019). Note that our prompts in the SoftP method select the prompt encoder trained

Method		Restarant		Laptop		Twitter	
		Accuracy	F1-score	Accuracy	F1-score	Accuracy	F1-score
FT	BERT-SPC	84.46	76.98	78.99	75.03	74.13	72.73
	SCAPT	83.39	74.53	77.17	73.23	74.64	73.91
PT	SentiPrompt*	84.39	75.53	79.17	<u>76.23</u>	75.64	74.88
	MRCOOL	85.49	**79.14**	78.12	75.78	-	-
Our	SynPrompt	84.17	76.45	78.28	75.36	74.88	73.73
	SynPrompt+AutoP	<u>85.67</u>	78.37	<u>80.79</u>	76.09	<u>75.84</u>	**74.37**
	SynPrompt+SoftP	**85.96**	<u>78.45</u>	**81.28**	**77.19**	**76.23**	<u>74.30</u>

Table 3: Experimental results (%) comparison on three publicly available datasets. We underline the second best performed baseline. The results of our rerunning version are marked with *.

Shot	Method	Restarant	Laptop	Twitter
1%	BERT-SPC	49.48	51.26	42.04
	SentiPrompt*	54.93 (+*5.45*)	54.06 (+*2.80*)	42.98 (+*0.94*)
	SynPrompt+AutoP	55.82 (+*6.34*)	55.12 (+*3.86*)	47.56 (+*5.52*)
	SynPrompt+SoftP	55.94 (+*6.46*)	54.53 (+*3.27*)	48.10 (+*6.06*)
2%	BERT-SPC	58.05	43.72	49.67
	SentiPrompt*	60.94 (+*2.49*)	49.93 (+*6.21*)	49.72 (+*0.05*)
	SynPrompt+AutoP	61.15 (+*3.10*)	53.06 (+*9.34*)	50.93 (+*1.30*)
	SynPrompt+SoftP	61.09 (+*3.04*)	53.96 (+*10.24*)	51.34 (+*1.67*)
4%	BERT-SPC	57.31	59.01	59.57
	SentiPrompt*	62.11 (+*4.80*)	61.65 (+*2.64*)	63.02 (+*3.45*)
	SynPrompt+AutoP	63.82 (+*6.51*)	62.65 (+*3.64*)	62.06 (+*2.49*)
	SynPrompt+SoftP	64.07 (+*6.76*)	60.64 (+*1.63*)	62.48 (+*2.91*)
8%	BERT-SPC	74.02	62.58	63.90
	SentiPrompt*	74.20 (+*0.18*)	66.67 (+*4.09*)	64.40 (+*0.50*)
	SynPrompt+AutoP	75.02 (+*1.00*)	66.92 (+*4.34*)	65.77 (+*1.87*)
	SynPrompt+SoftP	75.96 (+*1.94*)	67.03 (+*4.45*)	66.76 (+*2.86*)

Table 4: Experimental results (%) comparison of accuracy on three publicly available datasets through four levels of shot. The results of our rerunning version are marked with *.

by ourselves. Bert-base-uncased[2] was chosen as the main architecture for our model (n_{layers}=12, n_{heads}=12, n_{hidden}=768). The training batch size used is 16 for all models. The Adam optimizer is used for training, and the model is evaluated using two widely used metrics(accuracy and F1-score). The model is run three times with different seeds, and the average performance is reported. As for the number of the Prompt token, we set it to 3 by default for the following experiment.

5.3. Few-shot Setting

We tailored the dataset to a certain extent for few-shot learning experiments. We strictly define the size of the shot sample, which takes the same number of samples from each of the three categories. Experimentally, we randomly sample [1, 2, 4, 8] shots from each dataset for training, which were [1%,2%,4%,8%] of the complete training set. The statistical information of three datasets are shown in Table 2. Specifically, for each shot that has been set, we divide it into three equal parts corresponding to three label classes, which are grouped as the final train set. We don't do anything to the test set.

5.4. Baseline

We compare our proposed method based on two adaptive prompt frameworks with the following models on the ABSC subtask:

- Prompt-free baselines: **BERT-SPC** (Song et al., 2019), **SCAPT** (Li et al., 2021b).
- Prompt-based baselines: **SentiPrompt** (Li et al., 2021a), **MRCOOL** (Yang and Zhao, 2022).

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

”Prompt-free” means that only fine-tuning operations are performed in the model without using prompt learning. ”Prompt-based” means that the idea of prompt learning is used in the model.

5.5. Full-shot Learning Results

The results of full-shot learning on all three datasets across different models are reported in Table 3. Intuitively, our SynPrompt based on our proposed prompt frameworks achieves performance comparable to the best results before in terms of accuracy and f1-score, which demonstrates the effectiveness of our method in capturing more important prompt knowledge from syntactic relations. Notably, SynPrompt+SoftP achieves a significant improvement over the baseline models on the three datasets. We exceed the prior SOTA model on accuracy by +1.98%, +2.07%, and +1.43% on Res, Lap, and Twitter respectively. This improvement is attributed to SoftP's capability to automatically learn prompt word embeddings. Above results highlight the superiority of our proposed prompt-tuning methods for the ABSA tasks.

5.6. Few-shot Learning Results

By setting up few-shot learning above (§5.3), we experiment with our method on three public datasets, and results are reported in Table 4. Our methods consistently outperform the baselines especially when only 1-2% of training instances are available. Meanwhile, it is obvious that our approach outperforms the baseline approach on the Laptop dataset. After investigation, it is found that this dataset has fewer samples than the other two datasets, which further indicates that our method is stronger in the low-resource scenario. The findings strongly affirm the notion that a meticulously crafted prompt possesses immense potential in extracting the acquired knowledge within pretrained models, thereby resulting in enhanced performance in few-shot scenarios. In addition to that, it proved again that the prompt-based method outperforms fine-tuning, to a large extent.

5.7. Analysis of Key Variables

5.7.1. Size of Distance Threshold

The key to determining the prompt word based on syntactic distance is to determine the size of the distance threshold k. We test different sizes of thresholds with full data training on the Laptop dataset and present the results in Figure 5. It can be seen that ”$k = 4$” has the best accuracy, while ” $k = 5$” has the highest F1 score. Therefore, it is reasonable to assume that the size of the distance threshold k in SynPrompt should not be too long, probably because this would lead to the introduction of knowledge unrelated to aspect words.

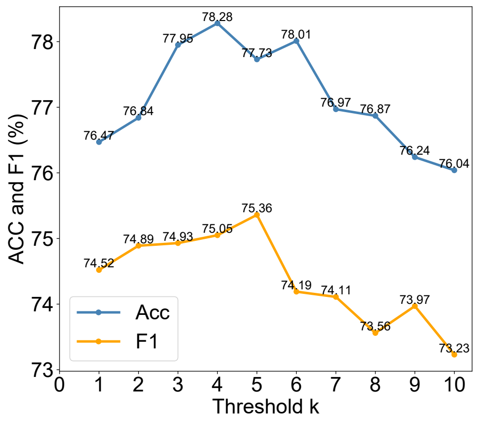


Figure 5: Effect of threshold k in SynPrompt. The results are produced under the full-shot setting on the Laptop dataset by vanilla SynPrompt.

Method	Template	Accuracy	F1-Score
AutoP	N=1	77.69	72.94
	N=2	77.82	73.05
	N=3	**78.14**	**73.29**
	N=4	77.95	72.91
	N=5	77.82	72.98
SoftP	N=1	77.98	72.98
	N=2	78.19	73.10
	N=3	**78.38**	73.19
	N=4	78.14	**73.29**
	N=5	78.17	73.07

Table 5: Effect of the number of prompt tokens N in the template. The results are produced under the full-shot setting on the Laptop dataset by vanilla AutoP and SoftP.

5.7.2. Number of Prompt Tokens

Given that the number of prompt tokens in AutoP and SoftP also plays a significant role in the performance of prompt engineering. Therefore we conducted experiments on the Laptop dataset under the full-shot setting and that result is shown in Figure 5. Specifically, we employed five hard prompt templates and five soft templates, varying the number of prompt tokens (denoted as N) for AutoP and SoftP, respectively. In the case of AutoP, we discovered that utilizing three prompt tokens was the optimal choice for this methodology. For SoftP, the model based on the prompt encoder showed the highest accuracy when using three prompt tokens, whereas four prompt tokens exhibited the best F1 score. These results underscore the significance of determining the most suitable length of the prompt tokens.

6. Conclusions

In this paper, we propose the SynPrompt to conduct specifically tailored prompts by fully exploiting the syntax information of the syntactic Dep.Tree. Additionally, we propose two prompt frameworks, which are adapted to our proposed SynPrompt and leverage pre-trained language models to improve the robustness and perception of SynPrompt. Our experimental results demonstrate that our approach is a more effective alternative to the traditional prompt-tuning method, especially few-shot scenarios, which also proves the rationality and explainability of our proposed method. Moreover, this work demonstrates that task-specific prompt engineering is critical.

This research work was supported by the National Natural Science Foundation of China (NSFC) (U19A2059) and the Sichuan Science and Technology Program(2022ZHCG0008). We appreciate all the authors for their fruitful discussions. In addition, our thanks to anonymous reviewers for their insightful comments and suggestions on improving the quality of this paper.

7. Limitations and Future Works

Despite the significant findings of this study, it is important to acknowledge several limitations. Firstly, the sample size is relatively small compared to other tasks, which may limit the generalizability of the results. Additionally, data collection is limited to a specific region, so further investigation is needed to determine the applicability of this approach to other tasks. Moreover, this study did not conduct case data analysis, which may restrict a comprehensive understanding of the impact of the method on the model. To address these limitations, future research could consider expanding the sample size and conducting semantic deep structure case analyses. Doing so, not only enhances the interpretability of this method but also facilitates the generalization of the method to broader directions.

We hope that our research will make a valuable contribution to the field of ABSA by offering fresh perspectives. Moving forward, we intend to extend our investigations into prompt engineering and verbalizers, while delving into prompt tuning that builds upon these concepts. Additionally, we aim to explore the untapped potential of pre-trained language models (PLMs) and evaluate their adaptability to specific tasks. By pursuing these avenues, we hope to advance the understanding and application of ABSA techniques.

8. Ethics Statement

Participant protection is of utmost importance in this study. All collected data will be anonymized to ensure participant confidentiality. Informed consent will be obtained from participants, who will have the freedom to withdraw from the study at any time. The research follows ethical guidelines, including integrity, respect, and fairness. Data security measures are in place to safeguard participant information, and data will be used solely for research purposes. Any concerns or complaints can be addressed through the established grievance mechanism. The study has undergone feasibility and ethical review, and participants' rights and privacy will be respected throughout the research process.

9. Bibliographical 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.

Li Dong, Furu Wei, Chuanqi Tan, Duyu Tang, Ming Zhou, and Ke Xu. 2014. Adaptive recursive neural network for target-dependent twitter sentiment classification. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics, ACL 2014, June 22-27, 2014, Baltimore, MD, USA, Volume 2: Short Papers*, pages 49–54. The Association for Computer Linguistics.

Tianhao Gao, Jun Fang, Hanyu Liu, Zhiyuan Liu, Chao Liu, Pengzhang Liu, Yongjun Bao, and Weipeng Yan. 2022. LEGO-ABSA: A prompt-based task assemblable unified generative framework for multi-task aspect-based sentiment analysis. In *Proceedings of the 29th International Conference on Computational Linguistics, COLING 2022, Gyeongju, Republic of Korea, October 12-17, 2022*, pages 7002–7012. International Committee on Computational Linguistics.

Tianyu Gao, Adam Fisch, and Danqi Chen. 2021. Making pre-trained language models better few-shot learners. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint*

Conference on Natural Language Processing, ACL/IJCNLP 2021, (Volume 1: Long Papers), Virtual Event, August 1-6, 2021, pages 3816–3830. Association for Computational Linguistics.

Long Jiang, Mo Yu, Ming Zhou, Xiaohua Liu, and Tiejun Zhao. 2011. Target-dependent twitter sentiment classification. In *The 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies, Proceedings of the Conference, 19-24 June, 2011, Portland, Oregon, USA*, pages 151–160. The Association for Computer Linguistics.

Akbar Karimi, Leonardo Rossi, and Andrea Prati. 2021. Improving BERT performance for aspect-based sentiment analysis. In *4th International Conference on Natural Language and Speech Processing, Trento, Italy, November 12-13, 2021*, pages 196–203. Association for Computational Linguistics.

Brian Lester, Rami Al-Rfou, and Noah Constant. 2021. The power of scale for parameter-efficient prompt tuning. 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 3045–3059. Association for Computational Linguistics.

Chengxi Li, Feiyu Gao, Jiajun Bu, Lu Xu, Xiang Chen, Yu Gu, Zirui Shao, Qi Zheng, Ningyu Zhang, Yongpan Wang, and Zhi Yu. 2021a. Sentiprompt: Sentiment knowledge enhanced prompt-tuning for aspect-based sentiment analysis. *CoRR*, abs/2109.08306.

Xiang Lisa Li and Percy Liang. 2021. Prefix-tuning: Optimizing continuous prompts for generation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, ACL/IJCNLP 2021, (Volume 1: Long Papers), Virtual Event, August 1-6, 2021*, pages 4582–4597. Association for Computational Linguistics.

Zhengyan Li, Yicheng Zou, Chong Zhang, Qi Zhang, and Zhongyu Wei. 2021b. Learning implicit sentiment in aspect-based sentiment analysis with supervised contrastive pre-training. 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 246–256. Association for Computational Linguistics.

Shuo Liang, Wei Wei, Xian-Ling Mao, Fei Wang, and Zhiyong He. 2022. Bisyn-gat+: Bi-syntax aware graph attention network for aspect-based sentiment analysis. In *Findings of the Association for Computational Linguistics: ACL 2022, Dublin, Ireland, May 22-27, 2022*, pages 1835–1848. Association for Computational Linguistics.

Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. 2023. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *ACM Comput. Surv.*, 55(9):195:1–195:35.

Xiao Liu, Yanan Zheng, Zhengxiao Du, Ming Ding, Yujie Qian, Zhilin Yang, and Jie Tang. 2021. GPT understands, too. *CoRR*, abs/2103.10385.

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.

Maria Pontiki, Dimitris Galanis, John Pavlopoulos, Harris 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.

Timo Schick and Hinrich Schütze. 2021. Exploiting cloze-questions for few-shot text classification and natural language inference. 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 255–269. Association for Computational Linguistics.

Taylor Shin, Yasaman Razeghi, Robert L. Logan IV, Eric Wallace, and Sameer Singh. 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 2020, Online, November 16-20, 2020*, pages 4222–4235. Association for Computational Linguistics.

Youwei Song, Jiahai Wang, Tao Jiang, Zhiyue Liu, and Yanghui Rao. 2019. Attentional encoder network for targeted sentiment classification. *CoRR*, abs/1902.09314.

Kai Wang, Weizhou Shen, Yunyi Yang, Xiaojun Quan, and Rui Wang. 2020. Relational graph attention network for aspect-based sentiment analysis. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 3229–3238. Association for Computational Linguistics.

Hu Xu, Bing Liu, Lei Shu, and Philip S. Yu. 2019. BERT post-training for review reading comprehension and aspect-based sentiment analysis. 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 2324–2335. Association for Computational Linguistics.

Yifei Yang and Hai Zhao. 2022. Aspect-based sentiment analysis as machine reading comprehension. In *Proceedings of the 29th International Conference on Computational Linguistics, COLING 2022, Gyeongju, Republic of Korea, October 12-17, 2022*, pages 2461–2471. International Committee on Computational Linguistics.

Wen Yin, Yi Xu, Cencen Liu, Dezhang Zheng, Qi Wang, and Chuanjie Liu. 2023. Prompt-oriented fine-tuning dual bert for aspect-based sentiment analysis. In *Artificial Neural Networks and Machine Learning – ICANN 2023*, pages 505–517, Cham. Springer Nature Switzerland.

Zheng Zhang, Zili Zhou, and Yanna Wang. 2022. SSEGCN: syntactic and semantic enhanced graph convolutional network for aspect-based sentiment analysis. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL 2022, Seattle, WA, United States, July 10-15, 2022*, pages 4916–4925. Association for Computational Linguistics.