Pretrained Language Encoders are Natural Tagging Frameworks for Aspect Sentiment Triplet Extraction

Anonymous ACL submission

Abstract

Aspect Sentiment Triplet Extraction (ASTE) aims to extract the spans of aspect, opinion, and their sentiment relations as sentiment triplets. Existing works usually formulate the span detection as a 1\textit{D} token tagging problem, and model the sentiment recognition with a 2\textit{D} tagging matrix of token pairs. Moreover, by leveraging the token representation of Pretrained Language Encoders (PLEs) like BERT, they can achieve better performance. However, they simply leverage PLEs as feature extractors to build their modules but never have a deep look at what specific knowledge does PLEs contain. In this paper, we argue that instead of further designing modules to capture the inductive bias of ASTE, PLEs themselves contain “enough” features for 1\textit{D} and 2\textit{D} tagging: (1) The token representation contains the contextualized meaning of token itself, so this level feature carries necessary information for 1\textit{D} tagging. (2) The attention matrix of different PLE layers can further capture multi-level linguistic knowledge existing in token pairs, which benefits 2\textit{D} tagging. (3) Furthermore, with simple transformations, these two features can also be easily converted to the 2\textit{D} tagging matrix and 1\textit{D} tagging sequence, respectively. That will further boost the tagging results. By doing so, PLEs can be natural tagging frameworks and achieve a new state of the art, which is verified by extensive experiments and deep analyses.

1 Introduction

Sentiment Analysis (Liu, 2012; Feldman, 2013) is an important Natural Language Understanding task (NLU) to identify the sentiment from review sentences, which has been widely studied in many fields, e.g., E-commerce (Shivaprasad and Shetty, 2017) and social media (Agarwal et al., 2011). Recently, Aspect-based Sentiment Analysis (Pontiki et al., 2014) tries to perform sentiment analysis at a fine-grained level, which comprises several subtasks, such as Aspect Term Extraction (Li et al., 2018), Aspect Opinion Extraction (Fan et al., 2019), and Aspect Sentiment Classification (Ruder et al., 2016). In order to provide a unified solution for these subtasks, Aspect Sentiment Triplet Extraction (ASTE) is proposed by (Peng et al., 2020) to extract sentiment triplets from review sentences, which contain all of the aspect terms, corresponding opinion spans, and their sentiment relations. For instance, given a review “The ambience was nice but the service wasn’t so great.”, the triplets of [ambience, nice, positive] and [service, wasn’t so great, negative] should be extracted.

To recognize the triplet elements, many efforts are devoted. Most of the existing works design various modules to detect the spans of aspect and opinion, as well as the sentiment relations of them, which can be divided into two categories: (1) (Peng et al., 2020; Chen et al., 2021a) conduct ASTE in multiple stages, which firstly extract aspect terms and opinion spans, and then combine the valid pairs of them and decide their sentiment relations. However, these methods suffer from error propagation: the wrongly extracted aspect and opinion spans can further influence the sentiment relation recognition. To address this issue, some works (Xu et al., 2020; Wu et al., 2020; Jing et al., 2021) formulate ASTE in an end-to-end manner, by designing tagging schemes (i.e., 1\textit{D} token level tagging scheme (Xu et al., 2020) and 2\textit{D} token pair tagging scheme
Specifically, they only simply use the token representation of PLEs as a backbone of their designed modules to capture the inductive bias of ASTE, such as the span information of aspect and opinion (Xu et al., 2021). However, we argue that is not the optimal way to leverage PLEs for ASTE, since the knowledge stored in them, i.e., token representation and attention matrix, is not fully used. As shown in Fig. 2, (1) The token representation contains the contextualized meaning of token itself, so this level feature carries necessary information to recognize aspect and opinion spans as a 1D tagging sequence (i.e., branch ① in Fig. 2). (2) The attention matrix of different layers in PLEs can capture multi-level linguistic knowledge existing in the token pairs. As (Jawahar et al., 2019) analyzed, the bottom layers focus more on phrase level syntactic information, and the top layers mainly capture semantic features. That means it contains effective features to recognize the sentiment relations between aspect and opinion spans with a 2D tagging matrix (i.e., branch ②). (3) Besides, the token representation and attention matrix can also be converted to 2D tagging matrix and 1D tagging sequence by some simple transformations (i.e., branches ③ and ④), so as to further boost the tagging results.

After these observations, we argue that PLEs themselves are naturally advanced tagging frameworks, due to the rich knowledge contained in both the token representation and attention matrix. All we need to do is converting them into the final tagging results. To this end, we propose SimpleTag to fully leverage the knowledge stored in both the feature sources of PLEs. Specifically, for the token representation, we use one branch to label the aspect and opinion spans with the token level tags of \(\{A, O, N\}\), where \(\{A, O\}\) means the token belongs to an Aspect or Opinion span; \(N\) means the token is Not one of the tokens of an aspect or opinion. Also, to detect the sentiment relations between the aspect and opinion, we leverage another branch to interact different token pairs in a multi-head selection manner (Bekoulis et al., 2018) with the token pair level tags \(\{Pos, Neu, Neg, N\}\). \(\{Pos, Neu, Neg\}\) means the token pair contains a Positive, Neutral or Negative sentiment relation if this token pair is from an aspect and opinion span respectively; \(N\) means the token pair does Not contain a sentiment relation.

For the attention matrix, we first use convolutional blocks to refine the attention matrix to obtain better token pair level features. Then, we also use two branches to model the refined attention matrix, which can further boost the aspect and opinion tagging, as well as the sentiment relation recognition between them. That is, one branch labels the token pairs of the same words (i.e., the diagonal of attention matrix) with the tags \(\{A, O, N\}\). Another branch assigns the token pair level features with the sentiment relation tags, i.e., \(\{Pos, Neu, Neg, N\}\). When the prediction is done, we apply a sample late fusion strategy to fuse the prediction logits to benefit our framework from both the knowledge of token representation and attention matrix. Finally, we use the fused result to decode sentiment triplets.

After conducting extensive experiments on four benchmarks (Peng et al., 2020; Xu et al., 2020), we demonstrate that our method can outperform the previous works and achieve a new state of the art. Moreover, the further analysis verifies that with our framework, the derived features of token representation and attention matrix are both important, where the task-specific features can be effectively distilled out from them and fully boost ASTE. We will release our code in the future.
To summarize, our contributions are as follows:

- We are the first to explicitly leverage the attention matrix derived from PLEs to access the token pair level knowledge for ASTE.
- We propose Simp1Tag, which is naturally derived from PLEs themselves. By leveraging both the token and token pair level features, we can fully mine the knowledge stored in PLEs to enhance both the 1D and 2D tagging results.
- The experimental results on four public benchmarks demonstrate that our method can achieve a new state of the art.

2 Related Works

Aspect Sentiment Triplet Extraction is proposed by (Peng et al., 2020), which aims to extract the triplets of all the aspect terms, opinion spans and the sentiment relations between them. To achieve this goal, many efforts are devoted. (Peng et al., 2020) proposes to extract the aspects and opinions at first, which will be combined into sentiment triplets later. (Chen et al., 2021a; Mao et al., 2021) transform ASTE task into a Machine Reading Comprehension (MRC) task to capture the connections among the subtasks of ASTE. (Huang et al., 2021) proposes a two-stage method to enhance the correlations between aspects and opinions. (Jian et al., 2021) proposes to regard the aspect and opinion terms as arguments of the expressed sentiment in a hierarchical reinforcement learning framework. (Xu et al., 2021) uses a span level approach to explicitly consider the interactions between the whole spans of aspects and opinions when predicting their sentiment relations. Besides, (Xu et al., 2020; Wu et al., 2020; Chen et al., 2021b; Jing et al., 2021) propose unified tagging schemes to extract sentiment triplets in one stage: (Xu et al., 2020) uses a token level tagging scheme, i.e., Position-aware Tagging Scheme, to extraction the sentiment triplets; (Wu et al., 2020; Chen et al., 2021b; Jing et al., 2021) use a token pair level tagging scheme, which results in a 2D tagging matrix. In addition, (Zhang et al., 2021b; Yan et al., 2021) both propose to extract the sentiment triplets via a generative way, where a sequence-to-sequence paradigm is used.

The difference between our framework and all the aforementioned works is that they only leverage the token representation of PLEs like BERT (Devlin et al., 2019) as the features to enhance the performance but ignore to leverage the rich linguistic knowledge hidden in its attention matrices of different layers (Jawahar et al., 2019; Clark et al., 2019). In contrast, after having a deep insight of PLEs, we treat them as a natural tagging framework to fully leverage the knowledge of both the feature sources, which is sample yet effective.

3 PLEs as Natural Tagging Frameworks

In this Section, we first describe the overall workflow of SimpleTag. Then, we elaborate on each component, i.e., Sentence Encoder, Tagging Layer (i.e., the branches ①-④ in Fig. 2), and Triplet Decoding Procedure.

3.1 Overall Workflow of SimpleTag

As show in Fig. 2, after the Pretrained Language Encoder (e.g., BERT) derives the token representation $H$ and attention matrix $A$. We fully and explicitly leverage ALL the knowledge of them in the Tagging Layer. (1) Branch ① and ④: the aspect and opinion span are predicted with the 1D token level tags $\{A, O, N\}$ by $T^{1D}_1 = \phi_1(H)$ and $T^{1D}_2 = \phi_2(A)$; (2) Branch ② and ③: the sentiment relation is predicted with the 2D token pair level tags $\{Pos, Neu, Neg, N\}$ by $T^{2D}_1 = \theta_1(H)$ and $T^{2D}_2 = \theta_2(A)$. Afterwards, we use a sample late fusion strategy to fuse the predicted logits of $T^{1D}$ and $T^{2D}$, so as to take advantage of both the token and token pair level knowledge of PLEs. Finally, the fused results are used to decode sentiment triplets.

3.2 Sentence Encoder

Our Sentence Encoder aims to fully access the knowledge in both the token representation and attention matrix of PLEs, where these rich linguistic features of token level and token pair level can be fully mined. Here we choose BERT as the representative of them, in order to keep consistent with most of the previous works.

Specifically, given one review sentence $\mathcal{S} = [w_1, w_2, ..., w_n]$, we first obtain its input embedding sequence. That is, $H_0 = [e_1, e_2, ..., e_n]$ ($e_i = w_i + p_i$), where $w_i$ and $p_i$ are the word embedding and position embedding of the $i$-th word. Then, the input embedding sequence is feed into BERT to obtain its token level representation and
token pair level attention matrix:
\[ H_i, A_i = BERT\_Layer_i(H_{i-1}), \ i \in [1, L], \]
where \( A_i \in \mathbb{R}^{h \times n \times n} \) is the derived \( h \) head attention matrix of \( i \)-th layers, and \( A^{1-L} \in \mathbb{R}^{(L+h) \times n \times n} \) is the stacked attention matrix of all BERT layers.

Afterwards, in order to better leverage the token pair level feature, we also leverage several convolution blocks to model this 2D matrix, which is a general way to refine the token pair level information and it has been applied in several NLP tasks like Incomplete Utterance Rewriting (Liu et al., 2020) and Document-level Relation Extraction (Zhang et al., 2021a).

Specifically, with the definition of one convolution block as:
\[ X'_{i-1} = \sigma(\text{conv}(X_{i-1})), \]
\[ X_i = \text{BatchNorm}(X'_{i-1}), \]
we conduct the convolution operation with the kernel size of \( 3 \times 3 \) and use ReLU as the activation function \( \sigma(\cdot) \). The channels of output are the same as the input. The refined process is as follows:
\[ X_0 = [A^{1-L}; R], \]
\[ X_i = \text{Conv\_Block}(X_{i-1}), \ i \in [1, C], \]
where \( R \in \mathbb{R}^{d_p \times n \times n} \) is the learnable parameters of relative position embeddings between token pairs. We use the final output \( X_C \) as our refined attention matrix \( A \), which is along with the token representation \( H = H_L \in \mathbb{R}^{n \times d} \) as the provided features for the following Tagging Layer.

We argue that this is a more effective way to leverage the pretrained knowledge in PLEs, since the attention matrix of different layers originally store the multi-level linguistic knowledge via the pretraining paradigm (Jawahar et al., 2019; Clark et al., 2019). In contrast, all the existing works only use the token representations of the last layer, which can result in losing task-specific features for ASTE.

### 3.3 Tagging Layer

After we get the token and token pair level features, i.e., \( H \) and \( A \), they are leveraged to predict the aspect and opinion spans as a 1D tagging sequence and the sentiment relations between them as a 2D tagging matrix.

#### 3.3.1 Aspect and Opinion Recognition

For aspect and opinion spans, we use the tags of \( \{A, O, N\} \) to label the these features.

Specifically, for \( H \) and \( A \), we implement \( T_1^{1D} = \phi_1(H) \) and \( T_2^{1D} = \phi_2(A) \) with two fully-connected layers to map them into the 1D tag sequences \( T_1^{2D}, T_2^{2D} \in \mathbb{R}^{n \times d} \), respectively. That is,
\[ T_1^{1D} = H W_1^{1D} + b_1^{1D}, \]
\[ T_2^{1D} = (\text{diagonal}(A)) W_2^{1D} + b_2^{1D}, \]
where the \( \text{diagonal}(\cdot) \) means taking the refined attention features between the same tokens. \( W_1^{1D}, W_2^{1D}, b_1^{1D} \) and \( b_2^{1D} \) are learnable parameters.

Finally, we use late fusion to sum both the prediction results, i.e., \( T^{1D} = T_1^{1D} + T_2^{1D} \) as the finally prediction to recognize aspect and opinion spans.

#### 3.3.2 Sentiment Relation Recognition

In the meanwhile, we also leverage \( H \) and \( A \) to recognize the sentiment relations between aspect and opinion by a 2D token pair level tagging scheme \( T^{2D} \in \mathbb{R}^{n \times n \times 4} \), where the classes belong to \( \{\text{Pos}, \text{Neu}, \text{Neg}, N\} \).

Specifically, to convert the token representation \( H \) to the 2D tagging matrix (i.e., \( T_1^{1D} = \theta_1(H) \)), we implement it with a multi-head selection mechanism (Bekoulis et al., 2018), where the t-th head is used to predict the t-th class between different token pairs. The detailed process for the t-th head is as follows:
\[ H_i^q = H_i W_1^q + b_1^q, i \in [1, 4], \]
\[ H_j^q = H_j W_2^q + b_2^q, \]
\[ T_{1,ij}^{2D,t} = H_i^q R_{i-j}(H_j^q)^T, \]
where \( T_1^{2D,t} \in \mathbb{R}^{n \times n} \) is the predicted tagging matrix for the t-th tag of \( \{\text{Pos}, \text{Neu}, \text{Neg}, N\} \). Here we denote \( T_1^{2D} \in \mathbb{R}^{n \times n \times 4} \) as the whole-class prediction from the token representation \( H \). \( R_{i-j} \) means the relative position embedding between the \( i \)-th and \( j \)-th token, where we implement it with the rotary position embedding (Su et al., 2021).

In addition, to convert the refined attention matrix to the 2D tagging matrix (i.e., \( T_2^{2D} = \theta_2(A) \)), we also implement this process with a fully-connected layer, which is as follows:
\[ T_2^{2D} = AW_2^{2D} + b_2^{2D} \in \mathbb{R}^{n \times n \times 4}, \]
Finally, we sum both prediction results to enhance the performance for sentiment relation recognition, i.e., \( T^{2D} = T_1^{2D} + T_2^{2D} \).
4 Experiment

4.1 Datasets

There are two versions of datasets for ASTE: ASTE-Data-V1 is released by (Peng et al., 2020) and ASTE-Data-V2 is released by (Xu et al., 2020). ASTE-Data-V1 does not contain cases where one opinion span is associated with multiple targets, but these cases are very common in the real world. ASTE-Data-V2 refines the V1 version with these additional missing triplets. Therefore, we mainly use ASTE-Data-V2 for our experiments, which is more general. Note that some works (Mao et al., 2021; Chen et al., 2021a) use ASTE-Data-V1 for the experiments. We also report the results of our method on ASTE-Data-V1 to fairly compare with them.

4.2 Evaluation Metrics

Following the existing works (Peng et al., 2020; Xu et al., 2020; Wu et al., 2020; Chen et al., 2021a), we use precision, recall, and F1 score as the metrics to evaluate the performance of ASTE. A correct triplet requires an exact match between the prediction of the aspect term, opinion span, and the sentiment polarity with the ground truth. Note that the F1 score takes into account both precision and recall, which can be regarded as a harmonic average of them. Therefore, we focus on the F1 score in the following experiments.

4.3 Implementation Details

The hyper-parameters in our experiment are tuned over the development set by grid search. We use \textit{bert-base-uncased} as our Sentence Encoder to be consistent with most of the previous works of ASTE. The learning rate of all the parameters is set to $5e-5$ with gradient clip of 1.0, where the Adam optimizer (Kingma and Ba, 2015) is used for model optimization with the cross entropy loss and a batch size of 16. Besides, the number of convolutional layers is selected from [2, 4, 6], and the dimension of the learnable relative position embeddings is set to 64. Our implementation is based on PyTorch (Paszke et al., 2019) and HuggingFace’s transformers library (Wolf et al., 2020) with a GeForce GTX 1080Ti GPU.

4.4 Compared Methods

Our method is compare with the mostly recent states of the arts on both ASTE-Data-V2 and V1 datasets, which are as follows:

- **JET**: (Xu et al., 2020) proposes to extract sentiment triplets by a position-aware tagging scheme.
- **GTS**: (Wu et al., 2020) uses a grid tagging scheme to extract sentiment triplets.
- **Span-ASTE**: (Xu et al., 2021) explicitly consider the interaction between the whole span of the aspect and opinion when predicting their sentiment.
- **(Jing et al., 2021)**: this work proposes a jointly optimized dual-encoder model for ABSA to boost the performance of ABSA tasks.
- **Dual-MRC**: proposes a dual-MRC framework to handle ASTE task, by jointly training two BERT-MRC models with parameter sharing.
- **BMRC**: (Chen et al., 2021a) proposes a bidirectional MRC framework to capture and utilize the associations among ASTE subtasks.
4.5 Overall Evaluation

As reported in Tab. 1, although these methods use Pretrained Models like BERT (Devlin et al., 2019) and BART (Lewis et al., 2020) in different ways to leverage its capability of deep language understanding, our method can consistently outperform all of them in both ASTE-Data-V2 and V1 datasets.

Specifically, for ASTE-Data-V2 (cf. the top of Tab. 1), compared to the recent state-of-the-art method Span-ASTE (Xu et al., 2021), although it use the token representation to create span level features, we can still outperform it by 1.94%, 2.04%, 1.15% and 1.62% on the four datasets respectively. In addition, to keep consistency and fairly compare with (Mao et al., 2021; Chen et al., 2021a), we also report the results of our method on ASTE-Data-V1, which are shown in the bottom of Tab. 1. Although these methods use Pretrained Models in a MRC way (Chen et al., 2021a; Mao et al., 2021) or generative way (Yan et al., 2021), as so to capture the discriminative features for ASTE, our method can also improve the performance by 0.9%, 0.18%, 1.62% and 2.65% on the four datasets respectively. The results on the two versions of datasets demonstrate that our method SimpleTag, which treats PLEs as natural tagging frameworks, can makes full and explicit use of the pretrained knowledge in PLEs and is more effective to tackle ASTE and can achieve a new state of the art.

5 Analysis and Discussion

In this part, we make deep analyses of SimpleTag from both qualitative and quantitative perspectives. The following experiments are based on ASTE-Data-V2.

5.1 Ablation Study

We firstly conduct various ablated experiments to analyze the contributions of different features in SimpleTag, where each branch (1-8) in Fig. 2 is removed as a variant.

As reported in Tab. 2, (1) When the branches from the attention matrix to 1D and 2D tagging are prohibited (i.e., w/o att. matrix), our framework can only get features from the token representation. That makes it degrade to a variant which is similar to (Wu et al., 2020), where their performance is at a same level: the performance of this variant averagely drops by 4.34 points. That demonstrates that the attention matrix of PLEs does contain much richer task-specific features for ASTE, which can provide more effective information than the modules proposed by the existing works. (2) If one of the branches rooted from the attention matrix is removed (i.e., w/o att. matrix → 1D / 2D tag), the performance of SimpleTag also declines by 2.15 and 3.27 points. That verifies the attention feature.
can boost both the aspect and opinion detection as well as sentiment relation recognition. (3) Besides, when we remove the convolution blocks (i.e., w/o convolution), its performance also drops by 2.1%, which means the convolution operation benefits for the refinement of attention matrix.

In addition, (4) When the two branches rooted from the token representation are all removed (i.e., w/o token rep.), the performance drop by 4.69%, which means the token representation also contains important information for ASTE. (5) When only one of these two branches is employed, the performance of SimpleTag can decline by 2.34% and 3.56%, which can draw a similar conclusion as (2).

Therefore, the above ablation study fully reveal the state-of-the-art performance can be achieved by simply converting the features of PLEs to 1D and 2D tagging matrix, which is the main contribution of this work.

Table 2: The ablation study of our method.

<table>
<thead>
<tr>
<th>Model</th>
<th>Res14 P</th>
<th>Res14 R</th>
<th>Res14 F1</th>
<th>Lap14 P</th>
<th>Lap14 R</th>
<th>Lap14 F1</th>
<th>Res15 P</th>
<th>Res15 R</th>
<th>Res15 F1</th>
<th>Res16 P</th>
<th>Res16 R</th>
<th>Res16 F1</th>
<th>Avg ΔF1</th>
</tr>
</thead>
<tbody>
<tr>
<td>SimpleTag-BERT</td>
<td>74.74</td>
<td>72.86</td>
<td>73.79</td>
<td>64.75</td>
<td>58.41</td>
<td>61.42</td>
<td>63.71</td>
<td>65.15</td>
<td>64.42</td>
<td>70.34</td>
<td>73.49</td>
<td>71.88</td>
<td>-</td>
</tr>
<tr>
<td>w/o att. matrix</td>
<td>69.92</td>
<td>70.84</td>
<td>70.38</td>
<td>61.59</td>
<td>53.05</td>
<td>57.00</td>
<td>56.46</td>
<td>61.24</td>
<td>58.75</td>
<td>64.84</td>
<td>71.54</td>
<td>68.03</td>
<td>-6.34</td>
</tr>
<tr>
<td>w/o att. matrix → 1D</td>
<td>70.55</td>
<td>71.54</td>
<td>71.04</td>
<td>60.74</td>
<td>58.04</td>
<td>59.36</td>
<td>62.19</td>
<td>62.06</td>
<td>62.13</td>
<td>71.03</td>
<td>69.79</td>
<td>70.40</td>
<td>-2.15</td>
</tr>
<tr>
<td>w/o att. matrix → 2D</td>
<td>70.33</td>
<td>71.04</td>
<td>70.68</td>
<td>59.57</td>
<td>56.93</td>
<td>58.22</td>
<td>60.41</td>
<td>60.41</td>
<td>60.41</td>
<td>66.67</td>
<td>71.73</td>
<td>69.11</td>
<td>-3.27</td>
</tr>
<tr>
<td>w/o convolution</td>
<td>72.05</td>
<td>72.05</td>
<td>72.05</td>
<td>59.54</td>
<td>57.67</td>
<td>58.59</td>
<td>62.61</td>
<td>60.41</td>
<td>61.49</td>
<td>70.44</td>
<td>71.54</td>
<td>70.99</td>
<td>-2.10</td>
</tr>
<tr>
<td>w/o token rep.</td>
<td>67.89</td>
<td>70.84</td>
<td>69.33</td>
<td>56.23</td>
<td>55.08</td>
<td>55.65</td>
<td>61.41</td>
<td>59.38</td>
<td>60.38</td>
<td>67.51</td>
<td>67.25</td>
<td>67.38</td>
<td>-4.69</td>
</tr>
<tr>
<td>w/o token rep. → 1D</td>
<td>72.11</td>
<td>71.24</td>
<td>71.68</td>
<td>56.91</td>
<td>57.12</td>
<td>57.01</td>
<td>62.02</td>
<td>63.30</td>
<td>62.65</td>
<td>69.75</td>
<td>71.93</td>
<td>70.83</td>
<td>-2.34</td>
</tr>
<tr>
<td>w/o token rep. → 2D</td>
<td>70.55</td>
<td>70.33</td>
<td>70.44</td>
<td>57.80</td>
<td>55.45</td>
<td>56.60</td>
<td>62.45</td>
<td>60.00</td>
<td>61.20</td>
<td>70.30</td>
<td>67.84</td>
<td>69.05</td>
<td>-3.56</td>
</tr>
</tbody>
</table>

Table 3: The effects (F1 scores) of attention matrices in different layers of BERT.

5.2 Effects of Attention in Different Layers

To further investigate the effects of attention matrices in different layers of BERT, we also remove its low layers (1-4), middle layers (5-8) and high layers (9-12) during model training and inference. As shown in Tab. 3: When we gradually remove the low, middle and high layers, the performance of SimpleTag averagely drops by 0.94%, 1.79% and 4.34%, which demonstrate that the layers of different levels do contain task-specific features and all of them contribute to the improvement. That corresponds to (Jawahar et al., 2019), which reveals the the low, middle and high layers of PLEs contain useful knowledge, i.e., the phrase, syntactic and semantic level information, respectively.

Table 4: The effects of SimpleTag with different Pretrained Models, where the experiments are based on Lap14.

<table>
<thead>
<tr>
<th>Model</th>
<th>P.</th>
<th>R.</th>
<th>F1</th>
</tr>
</thead>
<tbody>
<tr>
<td>BERT</td>
<td>64.75</td>
<td>58.41</td>
<td>61.42</td>
</tr>
<tr>
<td>XLNet</td>
<td>61.68</td>
<td>62.48</td>
<td>62.08</td>
</tr>
<tr>
<td>ELECTRA</td>
<td>62.87</td>
<td>62.29</td>
<td>62.58</td>
</tr>
</tbody>
</table>

5.3 Effects of Different Pretrained Models

Furthermore, instead of only applying our framework with the masked language model BERT, we also replace it with XLNet (Yang et al., 2019) and ELECTRA (Clark et al., 2020), where the former is permutation language model and the latter is trained in a generator-discriminator way. By doing so, we can verify the adaptability of our framework to other different pretraining paradigms.\(^1\) As shown in Tab. 4, compared with BERT-based Encoder, XLNET and ELECTRA can help our model achieve better performance with the F1 scores of 62.08 and 62.58. That demonstrates our model exhibits good robustness to different pretraining paradigms.

5.4 Qualitative Visualization of Attention

Besides, to demonstrate the knowledge stored in the attention matrices of different layers is beneficial to obtain informative and discriminative representations for ASTE, we also apply t-SNE

\(^1\)Please note that we use the base version for all of them.
(van der Maaten and Hinton, 2008) to these attention scores of different token pairs, and plot their 2-dimensional vectors in Fig. 4.\(^2\)

Specifically, we gradually concatenate the attention scores from the low to high layers. It is obvious that (1) Only using the bottom layers can easily tell the difference between the classes of \{A, O\} and \{Pos, Neu, Neg\} with a large margin. That indicates the bottom layers do capture some task-specific information existing in sentiment triplets. (2) By adding the attention scores of middle layers in BERT, we can observe that these features can further distinguish \(O\) from \(A\), and the sentiment class of \(Neg\) also begins to be distinguished from \(Pos\). (3) When the attention scores of all layers are used, it results in more compact clusters and clearer boundaries between different classes.

That suggests the features in different layers all contribute to the performance, which are helpful to decide the classes the token pairs belong to. Without any part of them can result in the situation of losing task-specific information.

### 5.5 Error Analysis

To guide the future works with deep insights, we also conduct error analysis on Res14 to investigate what wrong decisions are made by SimpleTag.

The main incorrect triplets are divided into four categories. Most of the errors decoded by our method are Span Detection Errors (43.8\%). For example, give the sentence “the french fries – with the kaimata dip were terrific!”, the aspect span comprised of multiple tokens (i.e., “the french fries – with kaimata dip”) is wrongly predicted as two independent aspect terms as “the french fries” and “kaimata dip”. This kind of mistake can be further solved by employing more effective convolution networks like U-Net (Ronneberger et al., 2015) to model the attention matrix, so as to detect span-level information (Xu et al., 2021).

Also, 35.6\% of the errors are introduced by misclassifying the sentiment relations. For the sentence “dessert was also to die for!”, our method predicts the sentiment as negative, which may be caused by the incorrect clue “die”. A solution for this mistake can be addressed by further adding phrase level information (Wang et al., 2021).

In addition, in the wrongly decoded triplets, 15.2\% of them are actually correct but they are not annotated in the ground truths, which means the datasets are not fully annotated. For “the sauce is excellent (very fresh) with dabs of real mozzarella.”, one of our predicted triplets is “(mozzarella, real, positive)”, which is correct but it is not annotated in the dataset.

Beside, some sentences require more powerful capability of language understanding to distinguish which triplets should not be extracted. For example, given “I came to fresh expecting a great meal, but all I got...”, our method wrongly predicts it with the triplet “(meal, great, positive)”. For this kind of error, more effective method is required to capture these semantic level information.

### 6 Conclusion

In this work, we propose SimpleTag, a tagging framework which is naturally derived from PLEs themselves with only simple transformations. By conducting various experiments, we demonstrate that both the token representation and attention matrix matter for ASTE, where fully leveraging these features can further make SimpleTag obtain state-of-the-art performance.
References


Kevin Clark, Minh-Thang Luong, Quoc V. Le, and Christopher D. Manning. 2020. ELECTRA: Pre-training text encoders as discriminators rather than generators. In ICLR.


Lianzhe Huang, Peiyi Wang, Sujuan 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.


