SpanMlt: A Span-based Multi-Task Learning Framework for Pair-wise Aspect and Opinion Terms Extraction

He Zhao [*], **Longtao Huang** [*†], **Rong Zhang**, **Quan Lu**, **Hui Xue**
Alibaba Group
{sicheng.zh,kaiyang.hlt,stone.zhangr,luquan.lq,hui.xueh}@alibaba-inc.com

Abstract

Aspect terms extraction and opinion terms extraction are two key problems of fine-grained Aspect Based Sentiment Analysis (ABSA). The aspect-opinion pairs can provide a global profile about a product or service for consumers and opinion mining systems. However, traditional methods can not directly output aspect-opinion pairs without given aspect terms or opinion terms. Although some recent co-extraction methods have been proposed to extract both terms jointly, they fail to extract them as pairs. To this end, this paper proposes an end-to-end method to solve the task of Pair-wise Aspect and Opinion Terms Extraction (PAOTE). Furthermore, this paper treats the problem from a perspective of joint term and relation extraction rather than under the sequence tagging formulation performed in most prior works. We propose a multi-task learning framework based on shared spans, where the terms are extracted under the supervision of span boundaries. Meanwhile, the pair-wise relations are jointly identified using the span representations. Extensive experiments show that our model consistently outperforms state-of-the-art methods.

1 Introduction

Fine-grained aspect-based sentiment analysis (ABSA) or opinion mining is a field of study that analyzes people's detailed insights towards a product or service. Aspect terms (AT) extraction and opinion terms (OT) extraction are two fundamental subtasks in ABSA (Pang and Lee., 2008; Liu, 2012). Aspect terms, also named as opinion targets, are the word sequences in the sentence describing attributes or features of the targets. Opinion terms, sometimes called opinion words, are those expressions carrying subjective attitudes. For example,

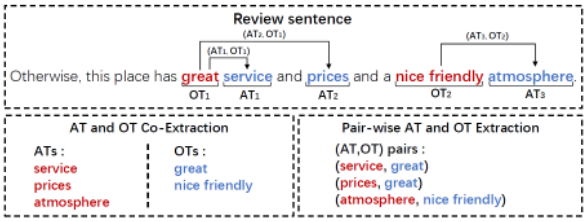


Figure 1: An example of the difference between co-extraction and pair extraction of AT and OT.

in the sentence "*Otherwise, this place has great service and prices and a nice friendly atmosphere*", the aspect terms are *service*, *prices* and *atmosphere*, and the opinion terms are *great* and *nice friendly*.

Recently, a new research focus, which aims at co-extracting the aspect and opinion terms (Wang et al., 2016, 2017; Li and Lam, 2017; Wang and Pan, 2018; Yu et al., 2019), has drawn increasing attention in both academia and industry. Such methods use joint models and have achieved great progress on both subtasks. However, the extracted AT and OT are not in pairs, and the corresponding relations between them are not well extracted. As the example sentence shown in Figure 1, (*service, great*), (*prices, great*) and (*atmosphere, nice friendly*) are three aspect-opinion pairs. In contrast, the co-extraction methods can only output the AT set {*service, prices, atmosphere*} and the OT set {*great, nice friendly*} jointly.

The aspect-opinion pairs can deploy more fine-grained sentiment analysis for review text and will benefit many downstream applications, such as opinion summarization and product profiling. By referring to the aspect-opinion pairs in a review sentence, customers can get a glimpse of the pros and cons of a product or service in a short time. Based on the promising results in previous AT and OT extraction, one possible solution for aspect-opinion pair extraction is to decouple the whole task into two subtasks. Firstly, all aspect terms need to be extracted from the sentences. Then, the OT cor-

[*]Both authors contributed equally to this research.
[†]Corresponding author

Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pages 3239–3248
July 5 - 10, 2020.

responding to each AT can be extracted using a Target-oriented Opinion Words Extraction (TOWE) method (Fan et al., 2019). Though this two-stage pipeline approach can extract aspect-opinion pairs, it will suffer from error propagation and the pairs extracting performance will rely heavily on the accuracy of AT extraction. To this end, an end-to-end method that can automatically extract AT and OT as pairs is essential for fine-grained sentiment analysis and opinion mining.

Considering the significance of the aspect-opinion pairs in review sentences, this paper targets at a new subtask for fine-grained ABSA, named PAOTE (Pair-wise Aspect and Opinion Terms Extraction). Given a review sentence, the objective of PAOTE is to extract all the (AT, OT) pairs. Different from the traditional co-extraction task of AT and OT, PAOTE outputs AT and OT in pairs while the co-extraction task only outputs them in separate sets as shown in Figure 1.

Most of the previous AT and OT extraction methods formulate the task as a sequence tagging problem (Wang et al., 2016, 2017; Wang and Pan, 2018; Yu et al., 2019), specifically using a 5-class tag set: {BA (beginning of aspect), IA (inside of aspect), BP (beginning of opinion), IP (inside of opinion), O (others)}. However, the sequence tagging methods suffer from a huge search space due to the compositionality of labels for extractive ABSA tasks, which has been proven in (Lee et al., 2017b; Hu et al., 2019). And as the example in Figure 1, the sequence tagging methods get into trouble when there exist *one-to-many* or *many-to-one* relations between AT and OT in the sentence.

In this paper, we propose a span-based multi-task framework to jointly extract both the AT/OT and the pair-wise relations. Motivated by prior works (Lee et al., 2017a; Luan et al., 2018), the proposed framework firstly learns word-level representations using a base encoder and then enumerates all possible spans on the input sentence. By sharing the generated span representations, the AT/OT can be extracted under the supervision of span boundaries and class labels. Meanwhile, the pair-wise relations can be identified by computing the span-span correspondence. We further design different encoder structures for the framework. To validate the effectiveness of our method, we conduct a serial of experiments based on public datasets. The comparison results show that the proposed framework can efficiently avoid the cascading errors between tasks and outperforms the state-of-the-art pipeline and joint methods.

In summary, the main contributions of this paper are concluded as follows:

1) We propose an end-to-end model for a new task PAOTE. To the best of our knowledge, it is the first end-to-end model that can jointly extract the AT/OT and the pair-wise relations between them.

2) We design a novel span-based multi-task neural network for PAOTE. It can overcome the drawbacks of sequence tagging methods by taking advantage of the span-level information. And the mutual impact between AT/OT and their pair-wise relations can be identified in this model.

3) We conduct extensive experiments and the results show that our proposed model outperforms the state-of-the-art methods.

2 Related Works

2.1 Aspect and Opinion Terms Extraction

For fine-grained ABSA, the aspect terms extraction and opinion terms extraction are two basic subtasks, which has been studied in numerous prior works (Hu and Liu, 2004; Popescu and Etzioni, 2005; Wu et al., 2009; Li et al., 2010; Qiu et al., 2011; Liu et al., 2012, 2013, 2015; Yin et al., 2016; Xu et al., 2019; Devlin et al., 2019). More recently, many works concentrate on co-extracting AT and OT using joint models. Most of the works treat the task as a sequence tagging problem. Wang et al. proposed a joint Recursive Neural Conditional Random Fields (RNCRF) model by using the dependency parse tree to capture dual-propagation among AT and OT (Wang et al., 2016). Then they extended their research and constructed a Recursive Neural Structural Correspondence Network (RNSCN) for cross-domain aspect and opinion terms co-extraction (Wang and Pan, 2018). Another outstanding work, Coupled Multi-Layer Attentions (CMLA) network, learns attentions for AT and OT (Wang et al., 2017). However, all these co-extraction methods do not consider the AT and OT as pairs.

For the pair-wise aspect and opinion terms extraction, an obvious solution is a two-stage pipeline strategy. The first stage is to extract aspect terms. Li et al. proposed a state-of-the-art model that can extract aspect terms by using the truncated history attention and the selective transformation network (Li et al., 2018). Then in the second stage, the target-oriented opinion terms can be extracted

with the given aspect terms. This subtask has been proposed in a recent work (Fan et al., 2019), where they develop a target-fused sequence tagging method. However, the opinion detection heavily depends on the extracted aspect accuracy, which suffers from error propagation. Our framework is the first to joint perform the two subtasks into an end-to-end model. Moreover, our method does not need any external lexicons or parsers and can effectively deal with multiple relations.

2.2 Joint Entity and Relation Extraction

Joint Entity and Relation Extraction (JERE), which aims to detect entity mentions and their semantic relations simultaneously in text, is an important task in information extraction. The earliest works mostly depend on feature engineering approaches (Kate and Mooney, 2010; Hoffmann et al., 2011; Li and Ji, 2014; Miwa and Sasaki, 2014). In recent studies, neural models for JERE have shown superior performance (Katiyar and Cardie, 2016; Zhang et al., 2017; Miwa and Bansal, 2016; Zheng et al., 2017). Moreover, neural multi-task learning has been shown effective in enhancing the interaction between entities and relations. In this paper, we adopt a JERE paradigm to solve the PAOTE task and develop a multi-task framework by extending previous unified setups (Luan et al., 2018) and end-to-end span-based models (Lee et al., 2017a, 2018).

3 Span-based Multi-task Framework

3.1 Problem Definition

Given an input sentence $S = \{w_1, w_2, ..., w_N\}$ of N words, the PAOTE task is to extract a set of all the aspect terms $AT = \{at_1, at_2, .., at_i\}$, a set of all the opinion terms $OT = \{ot_1, ot_2, ..., ot_j\}$ and a set of all the (AT, OT) pairs $P = \{(at_m, ot_n), ...\}$ from the sentence. Note that the $at_m \in AT$ and the $ot_n \in OT$ could be a single word or a phrase. Inspired by JERE methods, we process the task in a span-based *term-relation* joint extraction scheme rather than as a sequence tagging problem. Firstly, all possible spans $SP = \{s_1, s_2, ..., s_K\}$ are enumerated from the given sentence, where each span is a slice (up to a reasonable length l_s) of the input sentence. Based on the candidate spans, the outputs are two folds: 1) the term types T for all spans SP, aiming at the AT/OT recognition; 2) the pair-wise relation $\mathcal{R}$ for all pair of spans $SP \times SP$, aiming at the (AT, OT) pair identification. Formally, the two subtasks are defined as follows:

- **Term Recognition** is to assign a unique term label $\mathcal{T} \in \{A, O, null\}$ to each candidate span s_c, where A denotes $s_c \in AT$, O denotes $s_c \in OT$ and $null$ denotes that the span does not belong to AT or OT.

- **Pair-wise Relation Identification** is to assign a binary label $\mathcal{R} \in \{True, False\}$ to each ordered span pair (s_{c1}, s_{c2}). Note that the pair-wise relation is defined as a directed relation which always starts from an aspect term and points to an opinion term. So in this formulation, s_{c1} acts as AT and s_{c2} acts as OT. $True$ denotes that s_{c1} and s_{c2} are correctly associated.

3.2 Framework

The overall architecture of our span-based multi-task framework (**SpanMlt**) is shown in Figure 2. Given an input sentence, a base encoder is adopted to learn contextualized word representations. Then, a span generator is deployed to enumerate all possible spans, which are represented based on the hidden outputs of the base encoder. For the multi-task learning setup, the span representations are shared for two output scorers. The term scorer is to assign the term label with the highest score to each span. And the relation scorer is to evaluate the pair-wise correspondence between every two spans and assign a binary label to each span pair.

3.3 Span Generator

Given an input sentence $\{w_1, w_2, ..., w_N\}$, a span $s_i = \{w_{\text{START}(i)}, ..., w_{\text{END}(i)}\}$ is a single word or phrase with a starting index $\text{START}(i)$ and an ending index $\text{END}(i)$. And the maximum length of s_i is l_s:

$$1 \leq \text{START}(i) \leq \text{END}(i) \leq N \quad (1)$$

$$\text{END}(i) - \text{START}(i) < l_s \quad (2)$$

The span generator is a component enumerating all possible spans to generate the candidates for aspect or opinion terms. Then each span will be represented by using the contextualized word representations learned from various base encoders.

3.4 Base Encoders for Span Representations

Noting that SpanMlt is a general framework, we can potentially leverage any network as the encoder to learn word-level representations, which would be shared by higher-level modules. In this paper, we implement two different encoders. One is the

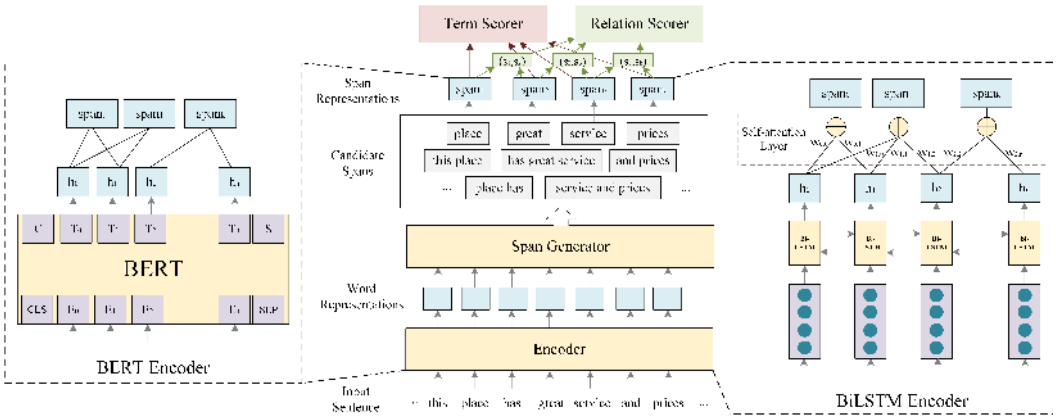


Figure 2: The overall architecture of the span-based multi-task framework, which alternatively takes a BERT structure or a BiLSTM structure as the base encoder to learn representations for input words and candidate spans.

BiLSTM with pre-trained word embeddings, which has been widely used in numerous neural-based models for NLP tasks. The other is BERT (Devlin et al., 2018), a pre-trained bidirectional transformer encoder which has achieved state-of-the-art performances across a variety of NLP tasks.

3.4.1 BiLSTM Encoder

For the BiLSTM encoder, the input vectors $\{x_1, x_2, ..., x_N\}$ are generated for the word sequence firstly. Motivated by (Lee et al., 2017a; Luan et al., 2018), two strategies are involved in building the vector representations: 1) pre-trained word embeddings and 1-dimension CNN over characters; 2) fixed ELMo embeddings. Then, a bidirectional LSTM network is used to encode each word x_t:

$$\boldsymbol{h}_t = [\overleftarrow{\text{LSTM}}(x_t); \overrightarrow{\text{LSTM}}(x_t)], t \in [1, N] \quad (3)$$

where $\boldsymbol{h}_t$ is the concatenated hidden output of BiLSTM.

To better learn vector representations combined with the syntactic head information for each candidate span, we further employ a self-attention layer over the word vectors in the span. Following previous works (Yang et al., 2016; Zhou et al., 2016), the attention is implemented with a feed forward neural network (FFNN):

$$u_t = \text{FFNN}_\alpha(\boldsymbol{h}_t, \theta_\alpha) \quad (4)$$

$$\alpha_{i,t} = \frac{exp(u_t)}{\sum_{k=\text{START}(i)}^{\text{END}(i)} exp(u_k)} \quad (5)$$

$$\hat{\boldsymbol{h}}_i = \sum_{k=\text{START}(i)}^{\text{END}(i)} \alpha_{i,t} \cdot u_t \quad (6)$$

where θ_α is the parameters for FFNN, and $\hat{\boldsymbol{h}}_i$ is a weighted sum of word vectors in span s_i. Therefore, based on the BiLSTM encoder, the final representation $\boldsymbol{p}_i$ for span s_i can be concatenated as:

$$\boldsymbol{p}_i = [\boldsymbol{h}_{\text{START}}(i); \boldsymbol{h}_{\text{END}}(i); \hat{\boldsymbol{h}}_i; \phi(i)] \quad (7)$$

where $\phi(i)$ is the feature vector encoding the size of the span s_i.

3.4.2 BERT Encoder

For the BERT encoder, the input sequence is generated by concatenating a [CLS] token, the original word sequence, and a [SEP] token. Each token is converted into an input vector $\boldsymbol{x}_t$ by summing the token, segment, and position embeddings. Assume $\text{BERT}(\cdot)$ is the base (or fine-tuned) BERT model. The hidden representation for each token can be obtained:

$$\boldsymbol{h}_t = \text{BERT}(\boldsymbol{x}_t) \quad (8)$$

Then the span vector representation $\boldsymbol{p}_i$ is directly generated by $\boldsymbol{h}_{\text{START}(i)}$ and $\boldsymbol{h}_{\text{END}(i)}$:

$$\boldsymbol{p}_i = [\boldsymbol{h}_{\text{START}}(i); \boldsymbol{h}_{\text{END}}(i)] \quad (9)$$

Unlike the BiLSTM encoder, we do not use the self-attention or the feature vector for the BERT encoder. Since the transformer of BERT has already utilized the attention mechanism and can learn sufficient contextualized information. And from our preliminary investigations and experiments, most complicated structures may damage the availability of BERT architecture and increase the training difficulty, which will be discussed in Section 4.

3.5 Objective

To construct the loss function for joint training, we use FFNNs over shared span representations to

compute the scores of how likely a span s_i has a term label $y_i^{\mathcal{T}}$, and how likely a span pair (s_i, s_j) has a relation label $y_{i,j}^{\mathcal{R}}$, respectively.

3.5.1 Term Scorer

For the term score, each span representation $\boldsymbol{p}_i$ is fed into an FFNN, and then is normalized with the softmax function to output the probability of the term label:

$$f_i^{\mathcal{T}} = \text{FFNN}_{\mathcal{T}}(\boldsymbol{p}_i, \theta_{\mathcal{T}}) \tag{10}$$

$$P(y_i^{\mathcal{T}}|s_i) = \text{Softmax}(f_i^{\mathcal{T}}) \tag{11}$$

Thus, the loss function for the term extraction subtask can be formulated using the span-level cross-entropy error between the predicted distribution $P(y_i^{\mathcal{T}}|s_i)$ and the gold distribution $P(y_i^{\mathcal{T}^*}|s_i)$:

$$Loss(\mathcal{T}) = -\sum_{i=1}^{k} P(y_i^{\mathcal{T}^*}|s_i) log(P(y_i^{\mathcal{T}}|s_i)) \tag{12}$$

3.5.2 Relation Scorer

For the pair-wise relation score between two spans (s_i, s_j), we first compute the probability that a span is in a relation:

$$f_i^{\mathcal{R}_s} = \text{FFNN}_{\mathcal{R}_s}(\boldsymbol{p}_i, \theta_{\mathcal{R}_s}) \tag{13}$$

In order to reduce the number of generated pairs, we sort the spans according to their scorers $f_i^{\mathcal{R}_s}$ and only the top-k spans are selected to be paired. Then, to measure the correspondence between two spans, the representation $\boldsymbol{p}_i$ for span s_i, the representation $\boldsymbol{p}_j$ for span s_j, and an element-wise multiplication $\boldsymbol{p}_i \odot \boldsymbol{p}_j$ are concatenated as the input of FFNN:

$$f_{i,j}^{\mathcal{R}} = \text{FFNN}_{\mathcal{R}}([\boldsymbol{p}_i; \boldsymbol{p}_j; \boldsymbol{p}_i \odot \boldsymbol{p}_j], \theta_{\mathcal{R}}) \tag{14}$$

The span scores and the correspondence score are summed and fed into the output softmax function:

$$P(y_{i,j}^{\mathcal{R}}|(s_i, s_j)) = \text{Softmax}(f_i^{\mathcal{R}_s} + f_j^{\mathcal{R}_s} + f_{i,j}^{\mathcal{R}}) \tag{15}$$

Thus, the loss function for the pair-wise relation extraction subtask can be formulated using the pair-level cross-entropy error between the predicted distribution $P(y_{i,j}^{\mathcal{R}}|(s_i, s_j))$ and the gold distribution $P(y_{i,j}^{\mathcal{R}^*}|(s_i, s_j))$:

$$Loss(\mathcal{R}) = -\sum_{i=1}^{k}\sum_{j=1}^{k} P(y_{i,j}^{\mathcal{R}^*}|(s_i, s_j)) log(P(y_{i,j}^{\mathcal{R}}|(s_i, s_j))) \tag{16}$$

Finally, losses from the term scorer and the relation scorer are combined as the training objective of the SpanMlt framework:

$$J(\theta) = \lambda_{\mathcal{T}} Loss(\mathcal{T}) + \lambda_{\mathcal{R}} Loss(\mathcal{R}) \tag{17}$$

where $\lambda_{\mathcal{T}}$ and $\lambda_{\mathcal{R}}$ are two hyper-parameters to balance the two tasks.

4 Experiments

4.1 Datasets

We evaluate our framework on two sets of public datasets, which are both in `LAPTOP` and `RESTAURANT` domains from Semeval 2014 Task 4, Semeval 2015 Task 12 and Semeval 2016 Task 5. One is provided by (Fan et al., 2019), where the AT and OT pairs are labeled. The other is provided by (Wang et al., 2017, 2016), where only the aspect terms and opinion terms are labeled.

4.2 Baselines

Since we are the first to study the joint extraction task of pair-wise AT and OT, there is no available end-to-end model in the literature to be compared. To better evaluate our method, we first compare the AT/OT extraction performances with several widely used sequence tagging models which are constructed by different encoder structures. Then we compare with three joint models, which have achieved state-of-the-art results in AT&OT co-extraction. To evaluate the extraction of (AT, OT) pairs, we further implement a pipeline approach HAST+TOWE. Moreover, since we formulate our problem as a joint term and relation extraction task, we also compare with a joint entity and relation extraction method JERE-MHS. These baselines are introduced as follows:

BiLSTM+CRF A sequence tagging method with a BiLSTM network built on top of pre-trained word embeddings, followed by a CRF output layer to perform BIO classification.

BERT+CRF A sequence tagging method based on a BERT encoder. The output hidden states of input words are taken as the features for CRF.

BERT+BiLSTM+CRF A sequence tagging method based on a BERT encoder. The output hidden states of input words are fed into a BiLSTM structure and then followed by an output CRF layer.

RNCRF A joint model of recursive neural network and CRF, proposed by (Wang et al., 2016) for single-domain AT and OT extraction.

CMLA A joint model of multi-layer attentions proposed by (Wang et al., 2017).

GMTCMLA A global inference model based on CMLA proposed by (Yu et al., 2019).

RNSCN A joint model proposed by (Wang and Pan, 2018) for cross-domain aspect and opinion terms extraction.

Models	14lap			14res			15res			16res		
	AT	OT	Pair	AT	OT	Pair	AT	OT	Pair	AT	OT	Pair
BiLSTM+CRF	69.80	64.96	-	78.03	75.13	-	66.27	64.70	-	70.43	73.33	-
BERT+CRF	56.38	50.14	-	54.37	48.41	-	57.01	45.95	-	55.83	49.38	-
BERT+BiLSTM+CRF	56.99	51.33	-	54.08	51.53	-	55.85	47.79	-	55.18	51.53	-
RNCRF	74.92	67.21	-	75.18	67.95	-	74.14	64.50	-	73.12	65.51	-
CMLA	75.57	66.27	-	76.08	66.32	-	78.31	66.15	-	76.84	65.73	-
RNSCN	73.71	75.89	-	82.12	81.67	-	71.02	69.78	-	75.11	72.18	-
HAST+TOWE (pipeline)	79.14	67.50	53.41	82.56	75.10	62.39	79.84	68.45	58.12	81.44	75.71	63.84
JERE-MHS	74.61	64.02	52.34	79.79	77.44	66.02	75.00	71.38	59.64	76.08	78.02	67.65
SpanMlt (ours)	**84.51**	**80.61**	**68.66**	**87.42**	**83.98**	**75.60**	**81.76**	**78.91**	**64.68**	**85.62**	**85.33**	**71.78**

Table 1: Main results (F1-score) for AT, OT and (AT, OT) pairs extraction on the four datasets from (Fan et al., 2019). State-of-the-art results are marked bold. SpanMlt with the best model setup achieves 15.25%, 9.58%, 5.04% and 4.13% absolute gains compared to the best pair extraction methods.

Models	14lap		14res		15res	
	AT	OT	AT	OT	AT	OT
RNCRF	78.42	79.44	84.93	84.11	67.47	67.62
CMLA	77.80	80.17	**85.29**	83.18	70.73	73.68
GMTCMLA	**78.69**	79.89	84.50	85.20	70.53	72.78
SpanMlt	77.87	**80.51**	85.24	**85.79**	**71.07**	**75.02**

Table 2: F1-scores for AT/OT extraction on the three datasets from (Wang et al., 2016, 2017).

HAST+TOWE (pipeline) A pipeline approach where the AT are first detected using a model proposed by (Li et al., 2018). Then given the predicted AT, the OT are extracted using a recent TOWE method (Fan et al., 2019). In this way, the pair-wise relation between AT and OT can be established.

JERE-MHS A model for joint entity-relation extraction, proposed by (Bekoulis et al., 2018). Although there are a number of complicated models for JERE, few works can simultaneously classify the entity types and the relation types. This method is the outstanding one which can be appropriate to solve our PAOTE task.

4.3 Hyperparameter Settings

For the BiLSTM encoder, we use the 300d GloVe word embeddings pre-trained on unlabeled data of 840 billion tokens[1]. We use a 3-layer BiLSTM with 100-dimension hidden states. The 8-dimensional char embeddings are randomly initialized. For the character CNN, the filter size is 50 with window sizes of 3, 4 and 5. The ELMo embeddings, pre-trained by a 3-layer BiLSTM with 1024 hidden states are fixed and not fine-tuned during the training stage. We use 0.4 dropout for the BiLSTMs and 0.5 dropout for the embeddings. The FFNNs are 50-dimensional with 2 hidden layers. The learning rate is set to be 0.005 for Adam optimizer.

For the BERT encoder, we use the pre-trained uncased BERT_{base} model[2], and run pre-training on `14lap` train set and on the sum of `14res`, `15res` and `16res` train set to get the domain-specific $\text{BERT}_{finetune}$ models, for `LAPTOP` and `RESTAURANT` respectively. The maximum sequence length is 512 with a batch size of 8. The FFNNs are 512-dimensional with a single hidden layer. The learning rate is set to 2e-5 for Adam optimizer.

The maximum length of generated spans is set to 8 and top 40% are candidate for pairs. $\lambda_{\mathcal{T}}$ and $\lambda_{\mathcal{R}}$ are both set to 1.0. We randomly split 10% of the train sets as dev sets for tuning the hyperparameters. Note that, all the baseline methods are implemented using their publicly released source codes. All the compared models are trained with best settings and the results for test sets are reported when it achieves the best performances on the dev sets.

4.4 Evaluation Metrics

We report F1 scores that measure the performance of our model and all the compared methods respectively for the three subtasks: AT extraction, OT extraction, and pair-wise relation extraction. An extracted AT or OT is regarded as a correct prediction when the boundaries of the span are identical to the ground-truth, and the term label is accurately assigned. An extracted pair-wise relation is correct only when both AT and OT are accurately identified and the relation label is accurately predicted.

4.5 Main Results

The main results are shown in Table 1. Our SpanMlt framework consistently achieves the best scores, both for the AT/OT extraction task and the pair-wise relation extraction task. For AT/OT extraction, the performance of sequence tagging methods is not satisfactory and the BERT-based models perform worst among all these methods. This suggests that BERT may not work well when the dataset for fine-tuning is small. The AT and OT co-extraction models perform much better than sequence tagging methods, indicating that the inter-

[1]https://nlp.stanford.edu/projects/glove/

[2]https://github.com/google-research/bert

Models	14lap			14res			15res			16res		
	AT	OT	Pair	AT	OT	Pair	AT	OT	Pair	AT	OT	Pair
SpanMlt-BERT$_{base}$	80.41	78.12	62.88	84.46	84.07	72.06	75.12	78.14	60.48	79.38	84.13	67.96
SpanMlt-BERT$_{finetune}$	80.78	79.71	65.75	84.26	84.11	72.72	77.71	78.47	61.06	80.95	84.92	69.58
SpanMlt-BiLSTM	81.30	77.58	64.41	83.02	83.42	73.80	80.14	76.48	59.91	82.44	83.87	67.72
- attention	78.69	76.83	62.88	82.55	81.22	71.97	79.48	75.12	59.22	81.90	83.50	67.21
- char embeddings	75.22	71.09	56.20	76.06	78.90	64.20	79.01	74.41	59.06	78.85	81.55	64.17
SpanMlt-BiLSTM-ELMo	84.51	80.61	68.66	87.42	83.98	75.60	81.76	78.91	64.68	85.62	85.33	71.78

Table 3: Comparisons for SpanMlt with different base encoders.

actions between AT and OT are significant for term extraction. However, all these joint models fail to associate AT and OT as pairs. For pair-wise AT/OT extraction, the HAST+TOWE pipeline method outperforms most other models on aspect detection, but the F1 scores of opinion extraction and pair extraction is much lower than that of SpanMlt, which is primarily due to the error propagation. Another joint entity and relation extraction method, namely JERE-MHS, performs worse than HAST for aspect extraction, but better than TOWE for opinion extraction.

To evaluate the efficacy of SpanMlt on separate AT or OT extraction more intuitively, we further compare with two state-of-the-art models on the larger public datasets from (Wang et al., 2016, 2017), which has no (AT, OT) pair labeled. Table 2 shows that our SpanMlt also achieves comparable results. The minor gap is because there exist some sentences only with AT or OT and without pair-wise relations in this dataset. Thus leads our method to fail to involve the impact of pair-wise relations.

4.6 Framework Analysis

Base Encoders. To further investigate the efficacy of different base encoders for our framework, namely, BiLSTM encoder and BERT encoder, we do experiments as shown in Table 3. The BiLSTM encoder with ELMo embeddings performs the best, which indicates the importance of initialized input embeddings. When using pre-trained Glove embeddings for BiLSTM encoder, the results are also satisfactory. An ablation study for the two key components, attention mechanism and char embeddings for BiLSTM encoder, suggests that both components are helpful for improving the performance. The BERT$_{base}$ encoder performs better in OT extraction but is inferior to the BiLSTM without ELMo in AT extraction. By using the BERT$_{finetune}$ model, the performance is improved, which indicates that introducing domain-specific information can help BERT to learn better contextualized word presentations. Figure 3 shows

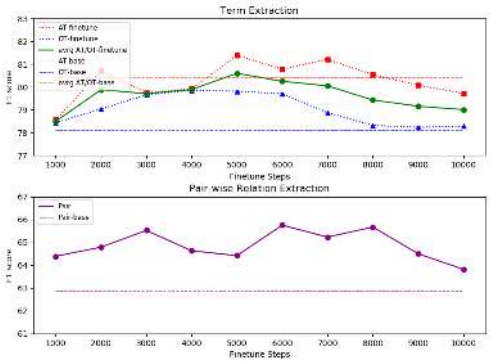


Figure 3: F1 curves on `14lap` dataset for the two tasks, using the base BERT model or fine-tuned BERT models with increasing training steps.

	AT	OT	Pair
Multi-task (SpanMlt)	84.51	80.61	68.66
Single-task Term	83.70	79.09	-
Single-task Relation	-	-	64.19

Table 4: Ablation study for multi-task learning on `14lap` test set.

F1 curves with increasing training steps for fine-tuning BERT on our `14lap` train set. We can see that the score first increases and achieves the highest at 5000-6000 steps. But then it decreases as the steps increasing. This result demonstrates that despite the domain-specific information is useful, too many steps on fine-tuning the pre-trained BERT models may not benefit the downstream tasks.

Multi-task Setup. We evaluate the effect of multi-task learning for the term extraction subtask and the pair-wise relation extraction subtask defined in our SpanMlt framework. Table 4 reports the F1 scores for an ablation study on `14lap` test set. It is observed that the performance improves when learning the two tasks jointly compared with each single task. In addition, to investigate the balance between the two subtasks for multi-task learning, we also draw the F1 curves when adjusting the loss weights $\lambda_{\mathcal{T}}$ and $\lambda_{\mathcal{R}}$, as shown in Figure 4. By varying $\lambda_{\mathcal{T}}/\lambda_{\mathcal{R}}$, we can see that the model attains the best performance at 1.00 for AT/OT extraction and 1.25 for pair-wise relation extraction. Nevertheless, our multi-task framework is relatively robust when varying the weight settings for the two subtasks.

Sentence	HAST+TOWE	SpanMlt
I've had it for about 2 months now and found no issues with software or updates.	(*software, no issues*)$_{\checkmark}$	(*software, no issues*)$_{\checkmark}$, (*updates, no issues*)$_{\checkmark}$
I seem to be having repeat problems as the Mother Board in this one is diagnosed as faulty, related to the graphics card.	(*Mother Board, problems*)$_{\times}$, (*graphics card, faulty*)$_{\checkmark}$	(*Mother Board, faulty*)$_{\checkmark}$, (*graphics card, faulty*)$_{\checkmark}$
Every time I log into the system after a few hours , there is this endlessly frustrating process that I have to go through.		(*system, frustrating*)$_{\times}$
My laptop with Windows 7 crashed and I did not want Windows 8.	(*Windows 8, crashed*)$_{\times}$	(*Windows 7, crashed*)$_{\checkmark}$

Table 5: Case study. The golden AT and OT in the sentences are colored as blue and red respectively. And the correct predictions are marked with ✓ and incorrect predictions are marked with ×.

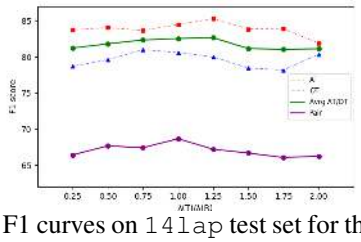
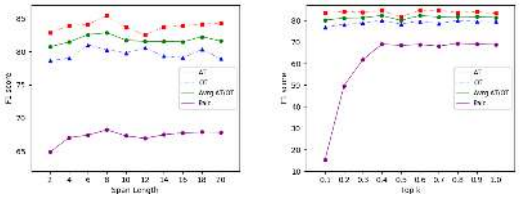


Figure 4: F1 curves on `14lap` test set for the two tasks using the best model setup when adjusting the loss balance, $\lambda_{\mathcal{T}}/\lambda_{\mathcal{R}}$.



(a) span length (b) top k

Figure 5: Effect of the maximum span length l_s and the top k of candidate spans with highest scores to be paired for our framework.

Parameter Sensitivity. Figure 5 shows F1 scores with different maximum span length l_s and different top k of candidate spans to generate pairs on `14lap` test set. We can see that F1 scores first increases as l_s becomes larger. But it slows the growth when the maximum span length is larger than 8. This indicates that too small l_s could not include all the useful words to generate the spans with accurate boundaries. Nevertheless, the extraction performance is not sensitive to maximum span length. For example, the difference between 8 and 20 are not statistically significant. For the number of candidate spans to generate pairs, top k, we can observe similar trends as that of span length. Too small k may cause that many correct AT and OT are not included in the candidate set, while large k will not improve extraction performance and may cost more training time.

4.7 Case Study

As mentioned previously, SpanMlt is able to identify *one-to-many* or *many-to-one* relationships between aspect and opinion terms. To verify that, we pick some examples from the test set of `14lap` and show the prediction results of SpanMlt and the pipeline approach HAST+TOWE, as presented in Table 5. In the first two cases, we can see that SpanMlt can correctly assign the same opinion term for two appositive aspect terms. While the pipeline method is less effective when dealing the *one-to-many* relations either by missing the correct AT (e.g. "*updates*") or assigning the incorrect OT (e.g. "*problems*"). Moreover, we find that our method may sometimes fail to recognize term boundaries (e.g., "*log into the system*" in case 3). There are also some bad cases due to the fact that our method fails to extract all pairs (e.g. "*Windows8*" and "*not want*" in case 4 are missed).

5 Conclusion

In this paper, we study a novel task Pair-wise Aspect and Opinion Terms Extraction (PAOTE). We treat this task as a joint term and relation extraction problem and develop a span-based multi-task learning framework (SpanMlt). Our framework can effectively learn contextualized information with various base encoders. Specifically, we try two different encoders (BiLSTM encoder and BERT encoder). Then a span generator enumerates all possible spans and each span is represented based on the outputs of the encoders. For joint optimizing the objectives of term extraction and pair-wise relation extraction, the two subtasks share the span representations and the losses are combined. The experimental results demonstrate that our SpanMlt significantly outperforms all the compared methods. For future works, we will explore pair-wise AT and OT extraction together with aspect category and sentiment polarity classification.

Acknowledgments

This research is supported in part by the National Natural Science Foundation of China under Grant 61702500.

References

Giannis Bekoulis, Johannes Deleu, Thomas Demeester, and Chris Develder. 2018. Joint entity recognition and relation extraction as a multi-head selection problem. *Expert Syst. Appl.*, 114:34–45.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *ArXiv*, abs/1810.04805.

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, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Zhifang Fan, Zhen Wu, Xin-Yu Dai, Shujian Huang, and Jiajun Chen. 2019. Target-oriented opinion words extraction with target-fused neural sequence labeling. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 2509–2518.

Raphael Hoffmann, Congle Zhang, Xiao Ling, Luke S. Zettlemoyer, and Daniel S. Weld. 2011. Knowledge-based weak supervision for information extraction of overlapping relations. In *ACL*.

Minghao Hu, Yuxing Peng, Zhen Huang, Dongsheng Li, and Yiwei Lv. 2019. Open-domain targeted sentiment analysis via span-based extraction and classification. *ArXiv*, abs/1906.03820.

Minqing Hu and Bing Liu. 2004. Mining and summarizing customer reviews. In *KDD*.

Rohit J. Kate and Raymond J. Mooney. 2010. Joint entity and relation extraction using card-pyramid parsing. In *CoNLL*.

Arzoo Katiyar and Claire Cardie. 2016. Investigating lstms for joint extraction of opinion entities and relations. In *ACL*.

Kenton Lee, Luheng He, Mike Lewis, and Luke S. Zettlemoyer. 2017a. End-to-end neural coreference resolution. *ArXiv*, abs/1707.07045.

Kenton Lee, Luheng He, and Luke S. Zettlemoyer. 2018. Higher-order coreference resolution with coarse-to-fine inference. In *NAACL-HLT*.

Kenton Lee, Tom Kwiatkowski, Ankur P. Parikh, and Dipanjan Das. 2017b. Learning recurrent span representations for extractive question answering. *ArXiv*, abs/1611.01436.

Fangtao Li, Chao Han, Minlie Huang, Xiaoyan Zhu, Yingju Xia, Shu Zhang, and Hao Yu. 2010. Structure-aware review mining and summarization. In *COLING*.

Qi Li and Heng Ji. 2014. Incremental joint extraction of entity mentions and relations. In *ACL*.

Xin Li, Lidong Bing, Piji Li, Wai Lam, and Zhimou Yang. 2018. Aspect term extraction with history attention and selective transformation. *ArXiv*, abs/1805.00760.

Xin Li and William W Y Lam. 2017. Deep multi-task learning for aspect term extraction with memory interaction. In *EMNLP*.

Bing Liu. 2012. Sentiment analysis and opinion mining. In *Synthesis Lectures on Human Language Technologies*.

Kang Liu, Heng Li Xu, Yang Liu, and Jun Zhao. 2013. Opinion target extraction using partially-supervised word alignment model. In *IJCAI*.

Kang Liu, Liheng Xu, and Jun Zhao. 2012. Opinion target extraction using word-based translation model. In *EMNLP-CoNLL*.

Pengfei Liu, Shafiq R. Joty, and Helen M. Meng. 2015. Fine-grained opinion mining with recurrent neural networks and word embeddings. In *EMNLP*.

Yi Luan, Luheng He, Mari Ostendorf, and Hannaneh Hajishirzi. 2018. Multi-task identification of entities, relations, and coreference for scientific knowledge graph construction. In *EMNLP*.

Makoto Miwa and Mohit Bansal. 2016. End-to-end relation extraction using lstms on sequences and tree structures. *ArXiv*, abs/1601.00770.

Makoto Miwa and Yutaka Sasaki. 2014. Modeling joint entity and relation extraction with table representation. In *EMNLP*.

Bo Pang and Lillian Lee. 2008. Opinion mining and sentiment analysis. In *Foundations and Trends in Information Retrieval*.

Ana-Maria Popescu and Oren Etzioni. 2005. Extracting product features and opinions from reviews. In *HLT/EMNLP*.

Guang Qiu, Bing Liu, Jiajun Bu, and Chun Chen. 2011. Opinion word expansion and target extraction through double propagation. *Computational Linguistics*, 37:9–27.

Wenya Wang and Sinno Jialin Pan. 2018. Recursive neural structural correspondence network for cross-domain aspect and opinion co-extraction. In *ACL*.

Wenya Wang, Sinno Jialin Pan, Daniel Dahlmeier, and Xiaokui Xiao. 2016. Recursive neural conditional random fields for aspect-based sentiment analysis. In *EMNLP*.

Wenya Wang, Sinno Jialin Pan, Daniel Dahlmeier, and Xiaokui Xiao. 2017. Coupled multi-layer attentions for co-extraction of aspect and opinion terms. In *AAAI*.

Yuanbin Wu, Qi Zhang, Xuanjing Huang, and Lide Wu. 2009. Phrase dependency parsing for opinion mining. In *EMNLP*.

Hu Xu, Bing Liu, Lei Shu, and Philip 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, Volume 1 (Long and Short Papers)*, pages 2324–2335, Minneapolis, Minnesota. Association for Computational Linguistics.

Zichao Yang, Diyi Yang, Chris Dyer, Xiaodong He, Alexander J. Smola, and Eduard H. Hovy. 2016. Hierarchical attention networks for document classification. In *HLT-NAACL*.

Yichun Yin, Furu Wei, Li Dong, Kaimeng Xu, Mingjie Zhang, and Mengchu Zhou. 2016. Unsupervised word and dependency path embeddings for aspect term extraction. In *IJCAI*.

Jianfei Yu, Jing Jiang, and Ruiping Xia. 2019. Global inference for aspect and opinion terms co-extraction based on multi-task neural networks. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 27:168–177.

Meishan Zhang, Yue Zhang, and Guohong Fu. 2017. End-to-end neural relation extraction with global optimization. In *EMNLP*.

Suncong Zheng, Feng Wang, Hongyun Bao, Yuexing Hao, Peng Zhou, and Bo Xu. 2017. Joint extraction of entities and relations based on a novel tagging scheme. *ArXiv*, abs/1706.05075.

Xinjie Zhou, Xiaojun Wan, and Jianguo Xiao. 2016. Attention-based lstm network for cross-lingual sentiment classification. In *EMNLP*.