Position-Aware Tagging for Aspect Sentiment Triplet Extraction

Lu Xu*[1,2], Hao Li*[1,3], Wei Lu[1], Lidong Bing[2]
[1] StatNLP Research Group, Singapore University of Technology and Design
[2] DAMO Academy, Alibaba Group [3]ByteDance
xu_lu@mymail.sutd.edu.sg, hao.li@bytedance.com
luwei@sutd.edu.sg, l.bing@alibaba-inc.com

Abstract

Aspect Sentiment Triplet Extraction (ASTE) is the task of extracting the triplets of target entities, their associated sentiment, and opinion spans explaining the reason for the sentiment. Existing research efforts mostly solve this problem using pipeline approaches, which break the triplet extraction process into several stages. Our observation is that the three elements within a triplet are highly related to each other, and this motivates us to build a joint model to extract such triplets using a sequence tagging approach. However, how to effectively design a tagging approach to extract the triplets that can capture the rich interactions among the elements is a challenging research question. In this work, we propose the first end-to-end model with a novel *position-aware* tagging scheme that is capable of jointly extracting the triplets. Our experimental results on several existing datasets show that jointly capturing elements in the triplet using our approach leads to improved performance over the existing approaches. We also conducted extensive experiments to investigate the model effectiveness and robustness[1].

1 Introduction

Designing effective algorithms that are capable of automatically performing sentiment analysis and opinion mining is a challenging and important task in the field of natural language processing (Pang and Lee, 2008; Liu, 2010; Ortigosa et al., 2014; Smailović et al., 2013; Li and Wu, 2010). Recently, Aspect-based Sentiment Analysis (Pontiki et al., 2014) or Targeted Sentiment Analysis (Mitchell et al., 2013) which focuses on extracting target phrases as well as the sentiment associated with each target, has been receiving much attention. In this work, we focus on a relatively new task – Aspect Sentiment Triplet Extraction (ASTE) proposed by Peng et al. (2019). Such a task is required to extract not only the targets and the sentiment mentioned above, but also the corresponding opinion spans expressing the sentiment for each target. Such three elements: a target, its sentiment and the corresponding opinion span, form a triplet to be extracted. Figure 1 presents an example sentence containing two targets in solid boxes. Each target is associated with a sentiment, where we use + to denote the positive polarity, 0 for neutral, and − for negative. Two opinion spans in dashed boxes are connected to their targets by arcs. Such opinion spans are important, since they largely explain the sentiment polarities for the corresponding targets (Qiu et al., 2011; Yang and Cardie, 2012).

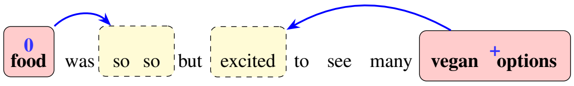


Figure 1: ASTE with targets in bold in solid squares, their associated sentiment on top, and opinion spans in dashed boxes. The arc indicates connection between a target and the corresponding opinion span.

This ASTE problem was basically untouched before, and the only existing work that we are aware of (Peng et al., 2019) employs a 2-stage pipeline approach. At the first stage, they employ a unified tagging scheme which fuses the target tag based on the $BIOES$[2] tagging scheme, and sentiment tag together. Under such a uni-

∗ Equal contribution. Lu Xu is under the Joint PhD Program between Alibaba and Singapore University of Technology and Design. The work was done when Hao Li was a PhD student in Singapore University of Technology and Design.

Accepted as a long paper in EMNLP 2020 (Conference on Empirical Methods in Natural Language Processing).

[1] We release our code at https://github.com/xuuuluuu/Position-Aware-Tagging-for-ASTE

[2] $BIOES$ is a common tagging scheme for sequence labeling tasks, and $BIOES$ denotes "begin, inside, outside, end and single" respectively.

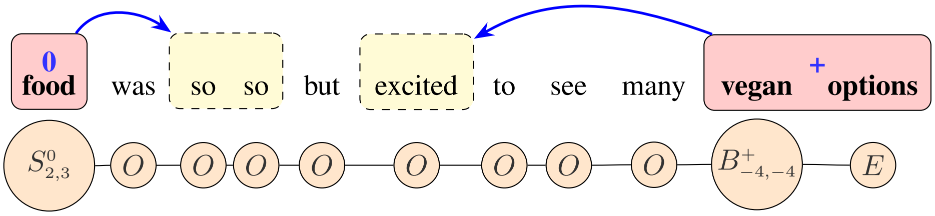


Figure 2: The position-aware tagging scheme for the example instance.

fied tagging scheme, they proposed methods based on Long Short-Term Memory networks (LSTM) (Hochreiter and Schmidhuber, 1997), Conditional Random Fields (CRF) (Lafferty et al., 2001) and Graph Convolutional Networks (GCN) (Kipf and Welling, 2017) to perform sequence labeling to extract targets with sentiment as well as opinion spans. At the second stage, they use a classifier based on Multi-Layer Perceptron (MLP) to pair each target (containing a sentiment label) with the corresponding opinion span to obtain all the valid triplets.

One important observation is that the three elements in a triplet are highly related to each other. Specifically, sentiment polarity is largely determined by an opinion span as well as the target and its context, and an opinion span also depends on the target phrase in terms of wording (e.g., an opinion span *"fresh"* usually describes food targets instead of service). Such an observation implies that jointly capturing the rich interaction among three elements in a triplet might be a more effective approach. However, the $BIOES$ tagging scheme on which the existing approaches based comes with a severe limitation for this task: such a tagging scheme without encoding any positional information fails to specify the connection between a target and its opinion span as well as the rich interactions among the three elements due to the limited expressiveness. Specifically, $BIOES$ uses the tag B or S to represent the beginning of a target. For example, in the example sentence in Figure 1, *"vegan"* should be labeled with B, but the tagging scheme does not contain any information to specify the position of its corresponding opinion *"excited"*. Using such a tagging scheme inevitably leads to an additional step to connect each target with an opinion span as the second stage in the pipeline approach. The skip-chain sequence models (Sutton and McCallum, 2004; Galley, 2006) are able to capture interactions between given input tokens which can be far away from each other. However, they are not suitable for the ASTE task where the positions of targets and opinion spans are not explicitly provided but need to be learned.

Motivated by the above observations, we present a novel approach that is capable of predicting the triplets jointly for ASTE. Specifically, we make the following contributions in this work:

- We present a novel position-aware tagging scheme that is capable of specifying the structural information for a triplet – the connection among the three elements by enriching the label semantics with more expressiveness, to address the above limitation.
- We propose a novel approach, **JET**, to Jointly Extract the Triplets based on our novel position-aware tagging scheme. Such an approach is capable of better capturing interactions among elements in a triplet by computing factorized features for the structural information in the ASTE task.
- Through extensive experiments, the results show that our joint approach **JET** outperforms baselines significantly.

2 Our Approach

Our objective is to design a model **JET** to extract the triplet of Target, Target Sentiment, and Opinion Span jointly. We first introduce the new position-aware tagging scheme, followed by the model architecture. We next present our simple LSTM-based neural architecture for learning feature representations, followed by our method to calculate factorized feature scores based on such feature representations for better capturing the interactions among elements in a triplet. Finally, we also discuss a variant of our model.

2.1 Position-Aware Tagging Scheme

To address the limitations mentioned above, we propose our position-aware tagging scheme by enriching expressiveness to incorporate position information for a target and the corresponding opinion span. Specifically, we extend the tag B and tag S in the $BIOES$ tagging scheme to new tags respectively:

$$B^{\epsilon}_{j,k}, S^{\epsilon}_{j,k}$$

where $B^{\epsilon}_{j,k}$ with the sub-tag[3] B still denotes the beginning of a target, and $S^{\epsilon}_{j,k}$ with the sub-tag S denotes a single-word target. Note that $\epsilon \in \{+, 0, -\}$ denotes the sentiment polarity for the target, and j, k indicate the position information which are the distances between the two ends of an opinion span and the starting position of a target respectively. Here, we use the term "*offset*" to denote such position information for convenience. We keep the other tags I, E, O as is. In a word, we use sub-tags $BIOES$ for encoding targets, ϵ for sentiment, and offsets for opinion spans under the new position-aware tagging scheme for the structural information.

For the example in Figure 1, under the proposed tagging scheme, the tagging result is given in Figure 2. The single-word target *"food"* is tagged with $S^{0}_{2,3}$, implying the sentiment polarity for this target is neutral (0). Furthermore, the positive offsets $2, 3$ indicate that its opinion span is on the right and has distances of 2 and 3 measured at the left and right ends respectively, (i.e., *"so so"*). The second target is *"vegan options"* with its first word tagged with $B^{+}_{-4,-4}$ and the last word tagged with E, implying the sentiment polarity is positive (+). Furthermore, the negative offsets $-4, -4$ indicate that the opinion span *"excited"* appears on the left of the target, and has distances of 4 and 4 measured at the left and right ends respectively, (i.e., *"vegan"*).

Our proposed position-aware tagging scheme has the following theoretical property:

Theorem 2.1. *There is a one-to-one correspondence between a tag sequence and a combination of aspect sentiment triplets within the sentence as long as the targets do not overlap with one another, and each has one corresponding opinion span.*[4]

Proof. For a given triplet, we can use the following process to construct the tag sequence. First note that the sub-tags of our proposed tags $B^{\epsilon}_{j,k}, I, O, E, S^{\epsilon}_{j,k}$, are B, I, O, E, S. The standard $BIOES$ tagset is capable of extracting all possible targets when they do not overlap with one another. Next, for each specified target, the position information j, k that specifies the position of its corresponding opinion span can be attached to the B (or S) tag, resulting in $B_{j,k}$ (or $S_{j,k}$). Note that the opinion span can be any span within the sentence when j, k are not constrained. Finally, we assign each extracted target its sentiment polarity ϵ by attaching it to the tag B (or S), resulting in $B^{\epsilon}_{j,k}$ (or $S^{\epsilon}_{j,k}$). This construction process is unique for each combination of triplets. Similarly, given a tag sequence, we can reverse the above process to recover the combination of triplets. □

We would like to highlight that our proposed position-aware tagging scheme is capable of handling some special cases where the previous approach is unable to. For example, in the sentence *"The salad is cheap with fresh salmon"*, there are two triplets, (*"salad"*, *"cheap with fresh salmon"*, positive)[5] and (*"salmon"*, *"fresh"*, positive). The previous approach such as (Peng et al., 2019), which was based on a different tagging scheme, will not be able to handle such a case where the two opinion spans overlap with one another.

2.2 Our JET Model

We design our novel **JET** model with CRF (Lafferty et al., 2001) and Semi-Markov CRF (Sarawagi and Cohen, 2004) based on our position-aware tagging scheme. Such a model is capable of encoding and factorizing both token-level features for targets and segment-level features for opinion spans.

Given a sentence $\mathbf{x}$ with length n, we aim to produce the desired output sequence $\mathbf{y}$ based on the position-aware tagging scheme. The probability of $\mathbf{y}$ is defined as:

$$p(\mathbf{y}|\mathbf{x}) = \frac{\exp\left(s(\mathbf{x}, \mathbf{y})\right)}{\sum_{\mathbf{y}' \in \mathbf{Y}_{\mathbf{x},M}} \exp(s(\mathbf{x}, \mathbf{y}'))} \quad (1)$$

where $s(\mathbf{x}, \mathbf{y})$ is a score function defined over the sentence $\mathbf{x}$ and the output structure $\mathbf{y}$, and $\mathbf{Y}_{\mathbf{x},M}$ represents all the possible sequences under our position-aware tagging scheme with the offset constraint M, indicating the maximum absolute value of an offset. The score $s(\mathbf{x}, \mathbf{y})$ is defined as:

$$s(\mathbf{x}, \mathbf{y}) = \sum_{i=0}^{n} \psi_{\bar{\mathbf{y}}_i, \bar{\mathbf{y}}_{i+1}} + \sum_{i=1}^{n} \Phi_{\mathbf{y}_i}(\mathbf{x}, i) \quad (2)$$

where $\bar{\mathbf{y}}_i \in \{B, I, O, E, S\}$ returns the sub-tag

[3] We define the sub-tags of $B^{\epsilon}_{j,k}, S^{\epsilon}_{j,k}$ as B and S respectively, and the sub-tags of I, O, E as themselves.

[4] See the Appendix for detailed statistics on how often this condition is satisfied.

[5] We use the format (target, opinion spans, sentiment).

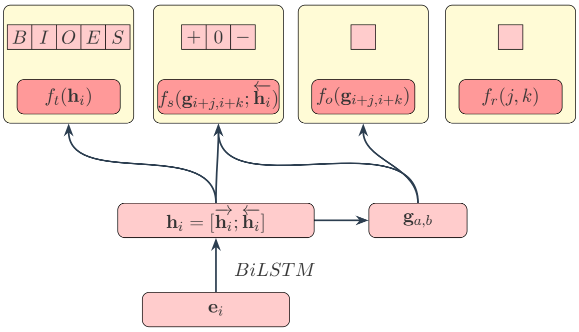

B I O E S
$+$ 0 $-$
$f_t(\mathbf{h}_i)$
$f_s(\mathbf{g}_{i+j,i+k}; \overleftarrow{\mathbf{h}}_i)$
$f_o(\mathbf{g}_{i+j,i+k})$
$f_r(j,k)$
$\mathbf{h}_i = [\overrightarrow{\mathbf{h}}_i; \overleftarrow{\mathbf{h}}_i]$
$\mathbf{g}_{a,b}$
$BiLSTM$
$\mathbf{e}_i$

Figure 3: Neural Module for Feature Score

of $\mathbf{y}_i$, $\psi_{\bar{\mathbf{y}}_i,\bar{\mathbf{y}}_{i+1}}$ represents the transition score: the weight of a "transition feature" – a feature defined over two adjacent sub-tags $\bar{\mathbf{y}}_i$ and $\bar{\mathbf{y}}_{i+1}$, and $\Phi_{\mathbf{y}_i}(\mathbf{x}, i)$ represents the factorized feature score with tag $\mathbf{y}_i$ at position i. In our model, the calculation of transition score $\psi_{\bar{\mathbf{y}}_i,\bar{\mathbf{y}}_{i+1}}$ is similar to the one in CRF[6]. For the factorized feature score $\Phi_{\mathbf{y}_i}(\mathbf{x}, i)$, we will explain computation details based on a simple LSTM-based neural network in the following two subsections. Such a factorized feature score is able to encode both token-level features as in standard CRF, segment-level features as in Semi-Markov CRF as well as the interaction among a target, its sentiment and an opinion span in a triplet.

2.2.1 Neural Module

We deploy a simple LSTM-based neural architecture for learning features. Given an input token sequence $\mathbf{x} = \{x_1, x_2, \cdots, x_n\}$ of length n, we first obtain the embedding sequence $\{\mathbf{e}_1, \mathbf{e}_2, \cdots, \mathbf{e}_n\}$. As illustrated in Figure 3, we then apply a bi-directional LSTM on the embedding sequence and obtain the hidden state $\mathbf{h}_i$ for each position i, which could be represented as:

$$\mathbf{h}_i = [\overrightarrow{\mathbf{h}}_i; \overleftarrow{\mathbf{h}}_i] \tag{3}$$

where $\overrightarrow{\mathbf{h}}_i$ and $\overleftarrow{\mathbf{h}}_i$ are the hidden states of the forward and backward LSTMs respectively.

Motivated by (Wang and Chang, 2016; Stern et al., 2017), we calculate the segment representation $\mathbf{g}_{a,b}$ for an opinion span with boundaries of a and b (both inclusive) as follows:

$$\mathbf{g}_{a,b} = [\overrightarrow{\mathbf{h}}_b - \overrightarrow{\mathbf{h}}_{a-1}; \overleftarrow{\mathbf{h}}_a - \overleftarrow{\mathbf{h}}_{b+1}] \tag{4}$$

where $\overrightarrow{\mathbf{h}}_0 = \mathbf{0}$, $\overleftarrow{\mathbf{h}}_{n+1} = \mathbf{0}$ and $1 \leq a \leq b \leq n$.

2.2.2 Factorized Feature Score

We explain how to compute the factorized feature scores (the second part of Equation 2) for the position-aware tagging scheme based on the neural architecture described above. Such factorized feature scores involve 4 types of scores, as illustrated in the solid boxes appearing in Figure 3 (top).

Basically, we calculate the factorized feature score for the tag $\mathbf{y}_i$ as follows:

$$\Phi_{\mathbf{y}_i}(\mathbf{x}, i) = f_t(\mathbf{h}_i)_{\bar{\mathbf{y}}_i} \tag{5}$$

where the linear layer f_t is used to calculate the score for local context for targets. Such a linear layer takes the hidden state $\mathbf{h}_i$ as the input and returns a vector of length 5, with each value in the vector indicating the score of the corresponding sub-tag among $BIOES$. The subscript $\bar{\mathbf{y}}_i$ indicates the index of such a sub-tag.

When $\mathbf{y}_i \in \{B^{\epsilon}_{j,k}, S^{\epsilon}_{j,k}\}$, we need to calculate 3 additional factorized feature scores for capturing structural information by adding them to the basic score as follows:

$$\Phi_{\mathbf{y}_i}(\mathbf{x}, i) \mathrel{+}= \tag{6}$$
$$f_s([\mathbf{g}_{i+j,i+k}; \overleftarrow{\mathbf{h}}_i])_\epsilon + f_o(\mathbf{g}_{i+j,i+k}) + f_r(j,k)$$

Note that the subscript of the variable $\mathbf{g}$ is represented as $i+j, i+k$ which are the absolute positions since j, k are the offsets. We explain such 3 additional factorized scores appearing in Equation 6.

- $f_s([\mathbf{g}_{i+j,i+k}; \overleftarrow{\mathbf{h}}_i])_\epsilon$ calculates the score for the sentiment. A linear layer f_s takes the concatenation of the segment representation $\mathbf{g}_{i+j,i+k}$ for an opinion span and the local context $\overleftarrow{\mathbf{h}}_i$ for a target, since we believe that the sentiment is mainly determined by the opinion span as well as the target phrase itself. Note that we only use the backward hidden state $\overleftarrow{\mathbf{h}}_i$ here, because the end position of a target is not available in the tag and the target phrase appears on the right of this position i. The linear layer f_s returns a vector of length 3, with each value representing the score of a certain polarity of $+, 0, -$. The subscript ϵ indicates the index of such a polarity.
- $f_o(\mathbf{g}_{i+j,i+k})$ is used to calculate a score for an opinion span. A linear layer f_o takes the segment representation $\mathbf{g}_{i+j,i+k}$ of an opinion span and returns one number representing the score of an opinion span.

[6] We calculate the transition parameters among five sub-tags $BIOES$ for targets.

Dataset	14Rest				14Lap				15Rest				16Rest			
	#S	# +	# 0	# -	#S	# +	# 0	# -	#S	# +	# 0	# -	#S	# +	# 0	# -
Train	1266	1692	166	480	906	817	126	517	605	783	25	205	857	1015	50	329
Dev	310	404	54	119	219	169	36	141	148	185	11	53	210	252	11	76
Test	492	773	66	155	328	364	63	116	322	317	25	143	326	407	29	78

Table 1: Statistics of 4 datasets. (#S denotes number of sentences, and # +, # 0, # - denote numbers of positive, neutral and negative triplets respectively.)

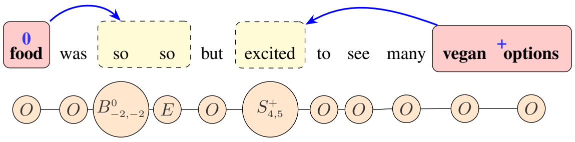


Figure 4: The gold tagging sequence of **JET**^{o} for the example sentence.

- $f_r(j,k)$ is used to calculate a score for offsets, since we believe the offset is an important feature. A linear layer f_r returns one number representing the score of offsets j,k which again are the distances between a target and two ends of the opinion span. Here, we introduce the offset embedding $\mathbf{w_r}$ randomly initialized for encoding different offsets. Specifically, we calculate the score as follows[7]:

  $$f_r(j,k) = W_r \mathbf{w_r}[\min{(j,k)}] + b_r \qquad (7)$$

  where W_r and b_r are learnable parameters.

2.3 One Target for Multiple Opinion Spans

The approach **JET** described above allows multiple targets to point to the same opinion span. One potential issue is that such an approach is not able to handle the case where one target is associated with multiple opinion spans. To remedy such an issue, we could swap a target and an opinion span to arrive at a new model as a model variant, since they are both text spans which are characterized by their boundaries. Specifically, in such a model variant, we still use the extended tags $B^{\epsilon}_{j,k}$ and $S^{\epsilon}_{j,k}$, where we use sub-tags $BIOES$ to encode an opinion span, the offsets j,k for the target and ϵ for the sentiment polarity. We use a similar procedure for the feature score calculation.

To differentiate with our first model, we name our first model as **JET**^{t} and such a model variant as **JET**^{o}. The superscripts t and o indicate the use of the sub-tags B and S to encode a target and an opinion span respectively. Figure 4 presents the gold tagging sequence of **JET**^{o}.

2.4 Training and Inference

The loss function $\mathcal{L}$ for the training data D is defined as:

$$\mathcal{L} = -\sum_{(\mathbf{x},\mathbf{y})\in D} \log p(\mathbf{y}|\mathbf{x}). \qquad (8)$$

The overall model is analogous to that of a neural CRF (Peng et al., 2009; Do et al., 2010; Lample et al., 2016); hence the inference and decoding follow standard marginal and MAP inference[8] procedures. For example, the prediction of $\mathbf{y}$ follows the Viterbi-like MAP inference procedure during decoding. Notice that the number of labels at each position under the position-aware tagging scheme is $O(M^2)$, since we need to compute segment representation for text spans of lengths within M. Hence, the time complexity for inference is $O(nM^2)$. When $M \ll n$ (empirically, we found n can be up to 80 in our datasets, and we set $M \in [2,6]$), this complexity is better than the existing work with complexity $O(n^2)$ (Peng et al., 2019).

3 Experiments

3.1 Data

We refine the dataset previously created by Peng et al. (2019)[9]. We call our refined dataset ASTE-Data-V2, and the original version as ASTE-Data-V1[10]. Note that ASTE-Data-V1 does not contain cases where one opinion span is associated with multiple targets. For example, there are two targets, "*service*" and "*atmosphere*", in the sentence "*Best service and atmosphere*". The opinion span "*Best*" is associated with such two targets, resulting in two triplets. However, we found that not all such triplets are explicitly annotated in ASTE-Data-V1. We refine the dataset with these additional missing triplets in our dataset ASTE-Data-

[7] We use $\min{(j,k)}$ since we care the offset between the starting positions of an opinion span and a target.

[8] See the Appendix for detailed algorithm.

[9] https://github.com/xuuuluuu/SemEval-Triplet-data

[10] We also report the results on ASTE-Data-V1 in the Appendix.

V2[11].

Table 1 presents the detailed statistics for 4 datasets.[12] 14Rest, 15Rest, 16Rest are the datasets of restaurant domain and 14Lap is of laptop domain. Such datasets were all created based on the datasets originally released by SemEval (Pontiki et al., 2014, 2015, 2016).

3.2 Baselines

Our **JET** approaches are compared with the following baselines using pipeline.

- **RINANTE+** (Peng et al., 2019) modifies **RINANTE** (Dai and Song, 2019) which is designed based on LSTM-CRF (Lample et al., 2016), to co-extract targets with sentiment, and opinion spans. Such an approach also fuses mined rules as weak supervision to capture dependency relations of words in a sentence at the first stage. At the second stage, it generates all the possible triplets and applies a classifier based on MLP on such triplets to determine if each triplet is valid or not.
- **CMLA+** (Peng et al., 2019) modifies **CMLA** (Wang et al., 2017) which leverages attention mechanism to capture dependencies among words, to co-extract targets with sentiment, and opinion spans at the first stage. At the second stage, it uses the same method to obtain all the valid triplets as **RINANTE+**.
- **Li-unified-R** (Peng et al., 2019) modifies the model (Li et al., 2019) to extract targets with sentiment, as well as opinion spans respectively based on a customized multi-layer LSTM neural architecture. At the second stage, it uses the same method to obtain all the valid triplets as **RINANTE+**.
- Peng et al. (2019) proposed an approach motivated by **Li-unified-R** to co-extract targets with sentiment, and opinion spans simultaneously. Such an approach also fuses GCN to capture dependency information to facilitate the co-extraction. At the second stage, it uses the same method to obtain all the valid triplets as **RINANTE+**.

3.3 Experimental Setup

Following the previous work (Peng et al., 2019), we use pre-trained 300d GloVe (Pennington et al., 2014) to initialize the word embeddings. We use 100 as the embedding size of $\mathbf{w_r}$ (offset embedding). We use the bi-directional LSTM with the hidden size 300. For experiments with contextualised representation, we adopt the pre-trained language model BERT (Devlin et al., 2019). Specifically, we use bert-as-service (Xiao, 2018) to generate the contextualized word embedding without fine-tuning. We use the representation from the last layer of the uncased version of BERT base model for our experiments.

Before training, we discard any instance from the training data that contains triplets with offset larger than M. We train our model for a maximal of 20 epochs using Adam (Kingma and Ba, 2014) as the optimizer with batch size 1 and dropout rate 0.5[13]. We select the best model parameters based on the best F_1 score on the development data and apply it to the test data for evaluation.

Following the previous works, we report the *precision* ($P.$), *recall* ($R.$) and F_1 scores for the correct triplets. Note that a correct triplet requires the boundary[14] of the target, the boundary of the opinion span, and the target sentiment polarity to be all correct at the same time.

3.4 Main Results

Table 2 presents the main results, where all the baselines as well as our models with different maximum offsets M are listed. In general, our joint models **JET**^{t} and **JET**^{o}, which are selected based on the best F_1 score on the dev set, are able to outperform the most competitive baseline of Peng et al. (2019) on the 4 datasets 14Rest, 15Rest, 16Rest, and 14Lap. Specifically, the best models selected from **JET**^{t} and **JET**^{o} outperform Peng et al. (2019) significantly[15] on 14Rest and 16Rest datasets with $p < 10^{-5}$ respectively. Such results imply that our joint models **JET**^{t} and **JET**^{o} are more capable of capturing interactions among the elements in triplets than those pipeline approaches. In addition, we observe a general trend from the results that the F_1 score increases as M increases on the 4 datasets when $M \leq 5$. We observe that the performance of **JET**^{t} and **JET**^{o} on the dev set of 14Lap drops when $M = 6$.

[11] We also remove triplets with sentiment originally labeled as "conflict" by SemEval.

[12] See the Appendix for more statistics.

[13] See the Appendix for experimental details. We use a different dropout rate 0.7 on the dataset 14Lap based on preliminary results since the domain is different from the other 3 datasets.

[14] We define a boundary as the beginning and ending positions of a text span.

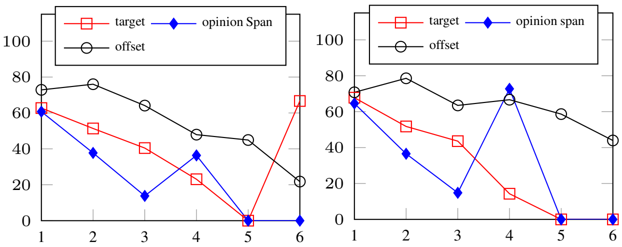
[15] We have conducted significance test using the bootstrap resampling method (Koehn, 2004).

Models	14Rest				14Lap				15Rest				16Rest			
	Dev F_1	$P.$	$R.$	F_1	Dev F_1	$P.$	$R.$	F_1	Dev F_1	$P.$	$R.$	F_1	Dev F_1	$P.$	$R.$	F_1
CMLA+	-	39.18	47.13	42.79	-	30.09	36.92	33.16	-	34.56	39.84	37.01	-	41.34	42.10	41.72
RINANTE+	-	31.42	39.38	34.95	-	21.71	18.66	20.07	-	29.88	30.06	29.97	-	25.68	22.30	23.87
Li-unified-R	-	41.04	67.35	51.00	-	40.56	44.28	42.34	-	44.72	51.39	47.82	-	37.33	54.51	44.31
Peng et al. (2019)	-	43.24	63.66	51.46	-	37.38	50.38	42.87	-	48.07	57.51	52.32	-	46.96	64.24	54.21
JET^t $(M = 2)$	45.67	72.46	32.29	44.68	35.69	57.39	24.31	34.15	42.34	64.81	28.87	39.94	43.27	68.75	38.52	49.38
JET^t $(M = 3)$	50.87	70.02	42.76	53.09	42.34	56.86	31.31	40.38	52.02	59.87	36.91	45.66	52.13	67.22	47.47	55.64
JET^t $(M = 4)$	50.31	69.67	47.38	56.41	45.90	48.77	32.78	39.21	52.50	64.50	40.82	50.00	57.69	64.64	47.67	54.87
JET^t $(M = 5)$	52.41	62.23	48.39	54.44	48.26	54.84	34.44	**42.31**	54.97	55.67	43.51	48.84	57.83	61.63	48.44	54.25
JET^t $(M = 6)$	53.14	66.76	49.09	**56.58**	47.68	52.00	35.91	42.48	55.06	59.77	42.27	**49.52**	58.45	63.59	50.97	**56.59**
JET^o $(M = 2)$	41.72	66.89	30.48	41.88	36.12	54.34	21.92	31.23	43.39	52.31	28.04	36.51	43.24	63.86	35.41	45.56
JET^o $(M = 3)$	49.41	65.29	41.45	50.71	41.95	58.89	31.12	40.72	48.72	58.28	34.85	43.61	53.36	72.40	47.47	57.34
JET^o $(M = 4)$	51.56	67.63	46.88	55.38	45.66	54.55	35.36	42.91	56.73	58.54	43.09	49.64	58.26	69.81	49.03	57.60
JET^o $(M = 5)$	53.35	71.49	47.18	56.85	45.83	55.98	35.36	**43.34**	59.57	61.39	40.00	48.44	55.92	66.06	49.61	56.67
JET^o $(M = 6)$	53.54	61.50	55.13	**58.14**	45.61	53.03	33.89	41.35	60.97	64.37	44.33	**52.50**	60.90	70.94	57.00	**63.21**
+ Contextualized Word Representation (BERT)																
JET^t $(M = 6)_{\text{+ BERT}}$	56.00	63.44	54.12	58.41	50.40	53.53	43.28	47.86	59.86	68.20	42.89	52.66	60.67	65.28	51.95	57.85
JET^o $(M = 6)_{\text{+ BERT}}$	56.89	70.56	55.94	62.40	48.84	55.39	47.33	51.04	64.78	64.45	51.96	57.53	63.75	70.42	58.37	63.83

Table 2: Main results on our refined dataset ASTE-Data-V2. The underlined scores indicate the best results on the dev set, and the highlighted scores are the corresponding test results. The experimental results on the previous released dataset ASTE-Data-V1 can be found in the Appendix.

For the dataset 14Rest, **JET**$^o(M = 6)$ achieves the best results on F_1 scores among all the **JET**^{o} models. Such a **JET**$^o(M = 6)$ model outperforms the strongest baseline Peng et al. (2019) by nearly 7 F_1 points. **JET**$^t(M = 6)$ also achieves a good performance with 56.58 in terms of F_1 score. Comparing results of our models to baselines, the reason why ours have better F_1 scores is that our models **JET**$^t(M \geq 4)$ and **JET**$^o(M \geq 4)$ both achieve improvements of more than 15 precision points, while we maintain acceptable recall scores. Similar patterns of results on the datasets 14Lap, 15Rest and 16Rest are observed, except that **JET**$^t(M = 5)$ and **JET**$^o(M = 5)$ achieves the best F_1 score on the dev set of 14Lap. Furthermore, we discover that the performance of both **JET**^{o} and **JET**^{t} on 14Rest and 16Rest datasets is better than on14Lap and 15Rest datasets. Such a behavior can be explained by the large distribution differences of positive, neutral and negative sentiment between the train and test set of the 14Rest and 16Rest datasets, shown in Table 1.

Furthermore, we also conduct additional experiments on our proposed model with the contextualized word representation BERT. Both **JET**^{t} $(M = 6)_{\text{+ BERT}}$ and **JET**^{o} $(M = 6)_{\text{+ BERT}}$ achieve new state-of-the-art performance on the four datasets.



(a) **JET**$^t(M = 6)_{\text{+ BERT}}$ (b) **JET**$^o(M = 6)_{\text{+ BERT}}$

Figure 5: $F_1(\%)$ scores (y-axis) of different lengths (x-axis) for targets, opinion spans and offsets on the dataset 14Rest.

4 Analysis

4.1 Robustness Analysis

We analyze the model robustness by assessing the performance on targets, opinion spans and offsets of different lengths for two models **JET**$^t(M = 6)_{\text{+ BERT}}$ and **JET**$^o(M = 6)_{\text{+ BERT}}$ on the four datasets. Figure 5 shows the results on the 14Rest dataset[16]. As we can see, **JET**$^o(M = 6)_{\text{+ BERT}}$ is able to better extract triplets with targets of lengths $\leq$ 3 than **JET**$^t(M = 6)_{\text{+ BERT}}$. Furthermore, **JET**$^o(M = 6)_{\text{+ BERT}}$ achieves a better F_1 score for triplets whose opinion spans are of length 1 and 4. However, **JET**$^o(M = 6)_{\text{+ BERT}}$ performs comparably to **JET**$^t(M = 6)_{\text{+ BERT}}$ for triplets whose opinion spans are of length 2 and 3. In addition, **JET**$^o(M = 6)_{\text{+ BERT}}$ is able to outperform **JET**$^t(M = 6)_{\text{+ BERT}}$ with offset of length 4 and above. We also observe that the performance drops when the lengths of targets, opinion spans and offsets are longer. This confirms that modeling the boundaries are harder when their lengths are longer. Similar patterns of results are observed on 14Lap, 15Rest, and 16Rest[17].

We also investigate the robustness on different evaluation methods, as presented in Figure 6. T

[16] See the Appendix for the statistics of accumulative percentage of different lengths for targets, opinion spans and offsets.

[17] See the Appendix for results on the other 3 datasets.

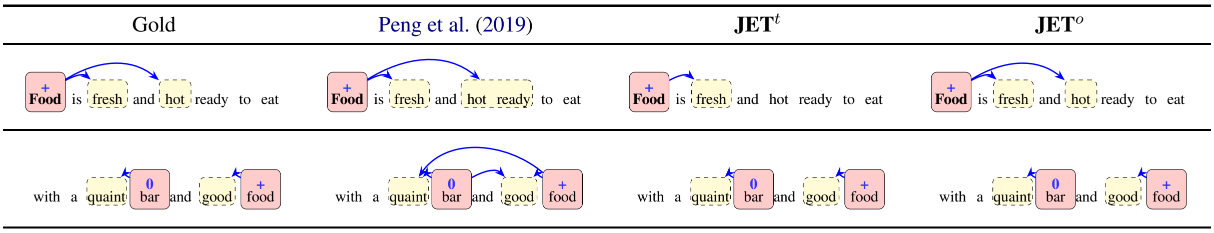
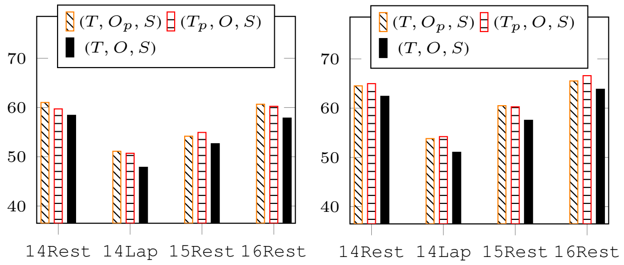

Gold
Peng et al. (2019)
JET^t
JET^o
+
Food is fresh and hot ready to eat
with a quaint bar and good food
0

Table 3: Qualitative Analysis



(a) **JET**$^t(M = 6)_{\text{+ BERT}}$ (b) **JET**$^o(M = 6)_{\text{+ BERT}}$

Figure 6: F_1 for different evaluation methods.

Model	14Rest		14Lap	
	JET^t	**JET**^{o}	**JET**^{t}	**JET**^{o}
$M = 6_{\text{+ BERT}}$	58.41	62.40	47.86	51.04
+char embedding	59.13	62.23	47.71	51.38
−offset features	55.36	61.24	44.16	49.58
−opinion span features	57.93	62.04	47.66	50.48
	15Rest		16Rest	
	JET^t	**JET**^{o}	**JET**^{t}	**JET**^{o}
$M = 6_{\text{+ BERT}}$	52.66	57.53	57.85	63.83
+char embedding	51.28	56.84	57.11	63.95
−offset features	48.74	53.68	52.83	61.72
−opinion span features	51.37	56.92	57.16	62.71

Table 4: Ablation Study (F_1)

(Target), O (Opinion Span) and S (Sentiment) are the elements to be evaluated. The subscript p on the right of an element in the legend denotes "partially correct". We define two boundaries to be partially correct if such two boundaries overlap. (T, O, S) is the evaluation method used for our main results. (T_p, O, S) requires the boundary of targets to be partially correct, and the boundary of opinion spans as well as the sentiment to be exactly correct. (T, O_p, S) requires the boundary of opinion spans to be partially correct, and the boundary of targets as well as the sentiment to be exactly correct. The results based on (T, O_p, S) yield higher improvements in terms of F_1 points than results based on (T_p, O, S), compared with (T, O, S) for **JET**$^t(M = 6)_{\text{+ BERT}}$ except on 15Rest. The results based on (T_p, O, S) yield higher F_1 improvements than results based on (T, O_p, S), compared with (T, O, S) for **JET**$^o(M = 6)_{\text{+ BERT}}$ except on 15Rest. Such a comparison shows the boundaries of opinion spans or target spans may be better captured when the sub-tags $BIOES$ are used to model the opinion or target explicitly.

4.2 Qualitative Analysis

To help us better understand the differences among these models, we present two example sentences selected from the test data as well as predictions by Peng et al. (2019), **JET**^{t} and **JET**^{o} in Table 3 [18]. As we can see, there exist 2 triplets in the gold data in the first example. Peng et al. (2019) predicts an incorrect opinion span *"hot ready"* in the second triplet. **JET**^{t} only predicts 1 triplet due to the model's limitation (**JET**^{t} is not able to handle the case of one target connecting to multiple opinion spans). **JET**^{o} is able to predict 2 triplets correctly. In the second example, the gold data contains two triplets. Peng et al. (2019) is able to correctly predict all the targets and opinion spans. However, it incorrectly connects each target to both two opinion spans. Our joint models **JET**^{t} and **JET**^{o} are both able to make the correct prediction.

4.3 Ablation Study

We also conduct an ablation study for **JET**$^t(M = 6)_{\text{+ BERT}}$ and **JET**$^o(M = 6)_{\text{+ BERT}}$ on dev set of the 4 datasets, presented in Table 4. "+char embedding" denotes concatenating character embedding into word representation. The results show that concatenating character embedding mostly has no much positive impact on the performance, which we believe is due to data sparsity. "−offset features" denotes removing $f_r(j, k)$ in the feature score calculation, Equation 6. F_1 scores drop more on the **JET**$^t(M = 6)_{\text{+ BERT}}$, this further confirms that modeling the opinion span is more difficult than target. "−opinion features" denotes removing $f_o(\mathbf{g}_{i+j,i+k})$ in the feature score calculation in Equation 6. F_1 scores drop consistently, implying the importance of such features for opinion spans.

4.4 Ensemble Analysis

As mentioned earlier, **JET**^{o} is proposed to overcome the limitation of **JET**^{t}, and vice versa. We

[18] See the Appendix for more examples.

Dataset	Model	$P.$	$R.$	F_1
14Rest	**JET**^{t}	63.44	54.12	58.41
	JET^o	70.56	55.94	62.40
	JET$^{o\to t}$	61.28	63.38	62.31
	JET$^{t\to o}$	61.10	63.98	**62.51**
14Lap	**JET**^{t}	53.53	43.28	47.86
	JET^o	55.39	47.33	51.04
	JET$^{o\to t}$	48.68	51.01	49.82
	JET$^{t\to o}$	49.57	53.22	**51.33**
15Rest	**JET**^{t}	68.20	42.89	52.66
	JET^o	64.45	51.96	57.53
	JET$^{o\to t}$	61.41	53.81	57.36
	JET$^{t\to o}$	61.75	55.26	**58.32**
16Rest	**JET**^{t}	65.28	51.95	57.85
	JET^o	70.42	58.37	**63.83**
	JET$^{o\to t}$	61.94	62.06	62.00
	JET$^{t\to o}$	62.50	63.23	62.86

Table 5: Results for Ensemble. We use the models **JET**^{t} and **JET**^{o} (with BERT, $M = 6$) as base models for building two ensemble models on 4 datasets.

believe that such two models complement each other. Hence, we propose two ensemble models **JET**$^{o\to t}$ and **JET**$^{t\to o}$ to properly merge the results produced by **JET**^{t} and **JET**^{o}. **JET**$^{o\to t}$ merges results of **JET**^{o} towards **JET**^{t} by adding distinct triplets from **JET**^{o} to **JET**^{t}, and analogously for **JET**$^{t\to o}$. We discuss how we build the ensemble models based on the two models **JET**^{t} and **JET**^{o} (with BERT, $M = 6$). First we call two triplets are *overlap* with one another if two targets overlap *and* any of their opinions overlap with one another. The ensemble model **JET**$^{o\to t}$ merges results from **JET**^{o} towards **JET**^{t}. Specifically, within the same instance, if a triplet produced by **JET**^{o} does not overlap with any triplet produced by **JET**^{t}, we augment the prediction space with such an additional triplet. After going through each triplet produced by **JET**^{o}, we regard the expanded predictions as the output of the ensemble model **JET**$^{o\to t}$. Similarly, we merge the result from **JET**^{t} towards **JET**^{o} to obtain the result for the ensemble model **JET**$^{t\to o}$.

We report results for ensemble models **JET**$^{o\to t}$ and **JET**$^{t\to o}$ presented in Table 5. As we can see, on `14Rest`, `14Lap` and `15Rest`, the ensemble model **JET**$^{t\to o}$ is able to achieve better F_1 score than **JET**^{t} and **JET**^{o}. However, such a simple ensemble approach appears to be less effective on `16Rest`. It is worth highlighting that the ensemble models have significant improvements in terms of recall score. Note that the recall score reflects the number of gold triplets extracted. Such improvement confirms our earlier hypothesis that the two models largely complement each other.

5 Related Work

ASTE is highly related to another research topic – Aspect Based Sentiment Analysis (ABSA) (Pontiki et al., 2014, 2016). Such a research topic focuses on identifying aspect categories, recognizing aspect targets as well as the associated sentiment. There exist a few tasks derived from ABSA. Target extraction (Chernyshevich, 2014; San Vicente et al., 2015; Yin et al., 2016; Lample et al., 2016; Li et al., 2018b; Ma et al., 2019) is a task that focuses on recognizing all the targets which are either aspect terms or named entities. Such a task is mostly regarded as a sequence labeling problem solvable by CRF-based methods. Aspect sentiment analysis or targeted sentiment analysis is another popular task. Such a task either refers to predicting sentiment polarity for a given target (Dong et al., 2014; Chen et al., 2017; Xue and Li, 2018; Wang and Lu, 2018; Wang et al., 2018; Li et al., 2018a; Peng et al., 2018; Xu et al., 2020) or joint extraction of targets as well as sentiment associated with each target (Mitchell et al., 2013; Zhang et al., 2015; Li and Lu, 2017; Ma et al., 2018; Li and Lu, 2019; Li et al., 2019). The former mostly relies on different neural networks such as self-attention (Liu and Zhang, 2017) or memory networks (Tang et al., 2016) to generate an opinion representation for a given target for further classification. The latter mostly regards the task as a sequence labeling problem by applying CRF-based approaches. Another related task – target and opinion span co-extraction (Qiu et al., 2011; Liu et al., 2013, 2014, 2015; Wang et al., 2017; Xu et al., 2018; Dai and Song, 2019) is also often regarded as a sequence labeling problem.

6 Conclusion

In this work, we propose a novel position-aware tagging scheme by enriching label expressiveness to address a limitation associated with existing works. Such a tagging scheme is able to specify the connection among three elements – a target, the target sentiment as well as an opinion span in an aspect sentiment triplet for the ASTE task. Based on the position-aware tagging scheme, we propose a novel approach **JET** that is capable of jointly extracting the aspect sentiment triplets. We also design factorized feature representations so as to effectively capture the interaction. We conduct extensive experiments and results show that our models outperform strong baselines significantly with detailed analysis. Future work includes finding applications of our novel tagging scheme in other tasks involving extracting triplets as well as extending our approach to support other tasks within sentiment analysis.

Acknowledgements

We would like to thank the anonymous reviewers for their helpful comments. This research is partially supported by Ministry of Education, Singapore, under its Academic Research Fund (AcRF) Tier 2 Programme (MOE AcRF Tier 2 Award No: MOE2017-T2-1-156). Any opinions, findings and conclusions or recommendations expressed in this material are those of the authors and do not reflect the views of the Ministry of Education, Singapore.

References

Heike Adel and Hinrich Schütze. 2017. Global normalization of convolutional neural networks for joint entity and relation classification. In *Proc. of EMNLP*.

Giannis Bekoulis, Johannes Deleu, Thomas Demeester, and Chris Develder. 2018. Adversarial training for multi-context joint entity and relation extraction. In *Proc. of EMNLP*.

Peng Chen, Zhongqian Sun, Lidong Bing, and Wei Yang. 2017. Recurrent attention network on memory for aspect sentiment analysis. In *Proc. of EMNLP*.

Maryna Chernyshevich. 2014. IHS r&d belarus: Cross-domain extraction of product features using CRF. In *Proc. of SemEval*.

Hongliang Dai and Yangqiu Song. 2019. Neural aspect and opinion term extraction with mined rules as weak supervision. In *Proc. of ACL*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proc. of NAACL*.

Trinh Do, Thierry Arti, et al. 2010. Neural conditional random fields. In *Proc. of AISTATS*.

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 *Proc. of ACL*.

Michel Galley. 2006. A skip-chain conditional random field for ranking meeting utterances by importance. In *Proc. of EMNLP*.

Ian Goodfellow, Jonathon Shlens, and Christian Szegedy. 2015. Explaining and harnessing adversarial examples. In *Proc. of ICLR*.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9:1735–80.

Yoon Kim. 2014. Convolutional neural networks for sentence classification. In *Proc. of EMNLP*.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. In *Proc. of ICLR*.

Thomas N Kipf and Max Welling. 2017. Semi-supervised classification with graph convolutional networks. In *Proc. of ICLR*.

Philipp Koehn. 2004. Statistical significance tests for machine translation evaluation. In *Proc. of EMNLP*.

John Lafferty, Andrew McCallum, and Fernando CN Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proc. of ICML*.

Guillaume Lample, Miguel Ballesteros, Sandeep Subramanian, Kazuya Kawakami, and Chris Dyer. 2016. Neural architectures for named entity recognition. In *Proc. of NAACL*.

Hao Li and Wei Lu. 2017. Learning latent sentiment scopes for entity-level sentiment analysis. In *Proc. of AAAI*.

Hao Li and Wei Lu. 2019. Learning explicit and implicit structures for targeted sentiment analysis. In *Proc. of EMNLP*.

Nan Li and Desheng Dash Wu. 2010. Using text mining and sentiment analysis for online forums hotspot detection and forecast. *Decision support systems*, 48(2).

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

Xin Li, Lidong Bing, Wai Lam, and Bei Shi. 2018a. Transformation networks for target-oriented sentiment classification. In *Proc. of ACL*.

Xin Li, Lidong Bing, Piji Li, and Wai Lam. 2019. A unified model for opinion target extraction and target sentiment prediction. In *Proc. of AAAI*.

Xin Li, Lidong Bing, Piji Li, Wai Lam, and Zhimou Yang. 2018b. Aspect term extraction with history attention and selective transformation. In *Proc. of IJCAI*.

Bing Liu. 2010. Sentiment analysis and subjectivity. *Handbook of natural language processing*.

Jiangming Liu and Yue Zhang. 2017. Attention modeling for targeted sentiment. In *Proc. of EACL*.

Kang Liu, Liheng Xu, and Jun Zhao. 2013. Syntactic patterns versus word alignment: Extracting opinion targets from online reviews. In *Proc. of ACL*.

Kang Liu, Liheng Xu, and Jun Zhao. 2014. Extracting opinion targets and opinion words from online reviews with graph co-ranking. In *Proc. of ACL*.

Pengfei Liu, Shafiq Joty, and Helen Meng. 2015. Fine–grained opinion mining with recurrent neural networks and word embeddings. In *Proc. of EMNLP*.

Dehong Ma, Sujian Li, and Houfeng Wang. 2018. Joint learning for targeted sentiment analysis. In *Proc. of EMNLP*.

Dehong Ma, Sujian Li, Fangzhao Wu, Xing Xie, and Houfeng Wang. 2019. Exploring sequence-to-sequence learning in aspect term extraction. In *Proc. of ACL*.

Margaret Mitchell, Jacqueline Aguilar, Theresa Wilson, and Benjamin Van Durme. 2013. Open domain targeted sentiment. In *Proc. of EMNLP*.

Makoto Miwa and Mohit Bansal. 2016. End-to-end relation extraction using LSTMs on sequences and tree structures. In *Proc. of ACL*.

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

Alvaro Ortigosa, José M Martín, and Rosa M Carro. 2014. Sentiment analysis in facebook and its application to e-learning. *Computers in Human Behavior*, 31.

Bo Pang and Lillian Lee. 2008. Opinion mining and sentiment analysis. *Foundations and trends in information retrieval*, 2(1-2).

Haiyun Peng, Yukun Ma, Yang Li, and Erik Cambria. 2018. Learning multi-grained aspect target sequence for chinese sentiment analysis. *Knowledge-Based Systems*, 148:167–176.

Haiyun Peng, Lu Xu, Lidong Bing, Fei Huang, Wei Lu, and Luo Si. 2019. Knowing what, how and why: A near complete solution for aspect-based sentiment analysis. In *Proc. of AAAI*.

Jian Peng, Liefeng Bo, and Jinbo Xu. 2009. Conditional neural fields. In *Proc. of NIPS*.

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. Glove: Global vectors for word representation. In *Proc. of EMNLP*.

Maria Pontiki, Dimitris Galanis, Haris Papageorgiou, Ion Androutsopoulos, Suresh Manandhar, Mohammed AL-Smadi, Mahmoud Al-Ayyoub, Yanyan Zhao, Bing Qin, Orphée De Clercq, et al. 2016. Semeval-2016 task 5: Aspect based sentiment analysis. In *Proc. of SemEval*.

Maria Pontiki, Dimitris Galanis, Haris Papageorgiou, Suresh Manandhar, and Ion Androutsopoulos. 2015. SemEval-2015 task 12: Aspect based sentiment analysis. In *Proc. of SemEval*.

Maria Pontiki, Dimitris Galanis, John Pavlopoulos, Harris Papageorgiou, Ion Androutsopoulos, and Suresh Manandhar. 2014. Semeval-2014 task 4: Aspect based sentiment analysis. In *Proc. of SemEval*.

Guang Qiu, Bing Liu, Jiajun Bu, and Chun Chen. 2011. Opinion word expansion and target extraction through double propagation. *Computational Linguistics*, 37(1):9–27.

Iñaki San Vicente, Xabier Saralegi, and Rodrigo Agerri. 2015. EliXa: A modular and flexible ABSA platform. In *Proc. of SemEval*.

Sunita Sarawagi and William W Cohen. 2004. Semi–markov conditional random fields for information extraction. In *Proc. of NIPS*.

Jasmina Smailović, Miha Grčar, Nada Lavrač, and Martin Žnidaršič. 2013. Predictive sentiment analysis of tweets: A stock market application. In *Human-Computer Interaction and Knowledge Discovery in Complex, Unstructured, Big Data*. Springer.

Mitchell Stern, Jacob Andreas, and Dan Klein. 2017. A minimal span-based neural constituency parser. In *Proc. of ACL*.

Charles Sutton and Andrew McCallum. 2004. Collective segmentation and labeling of distant entities in information extraction. Technical report, MASSACHUSETTS UNIV AMHERST DEPT OF COMPUTER SCIENCE.

Duyu Tang, Bing Qin, and Ting Liu. 2016. Aspect level sentiment classification with deep memory network. In *Proc. of EMNLP*.

Bailin Wang and Wei Lu. 2018. Learning latent opinions for aspect-level sentiment classification. In *Proc. of AAAI*.

Shuai Wang, Sahisnu Mazumder, Bing Liu, Mianwei Zhou, and Yi Chang. 2018. Target-sensitive memory networks for aspect sentiment classification. In *Proc. of ACL*.

Wenhui Wang and Baobao Chang. 2016. Graph-based dependency parsing with bidirectional lstm. In *Proc. of ACL*.

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

Han Xiao. 2018. bert-as-service. `https://github.com/hanxiao/bert-as-service`.

Hu Xu, Bing Liu, Lei Shu, and Philip S. Yu. 2018. Double embeddings and cnn-based sequence labeling for aspect extraction. In *Proc. of ACL*.

Lu Xu, Lidong Bing, Wei Lu, and Fei Huang. 2020. Aspect sentiment classification with aspect-specific opinion spans. In *Proc. of EMNLP*.

Wei Xue and Tao Li. 2018. Aspect based sentiment analysis with gated convolutional networks. In *Proc. of ACL*.

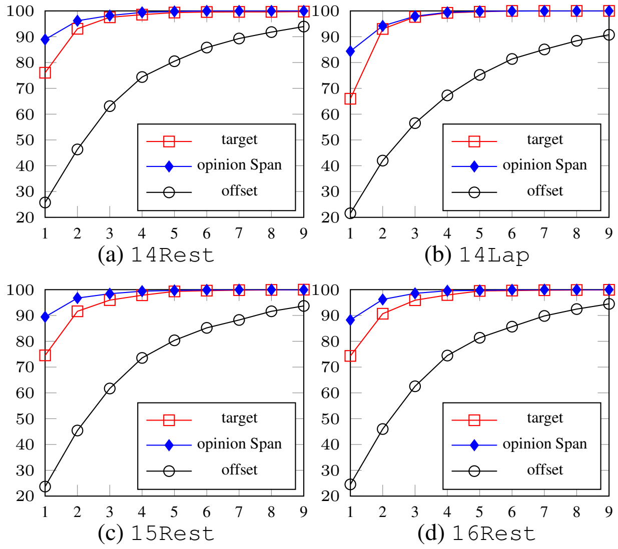
Bishan Yang and Claire Cardie. 2012. Extracting opinion expressions with semi-Markov conditional random fields. In *Proc. of EMNLP*.

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

Meishan Zhang, Yue Zhang, and Duy-Tin Vo. 2015. Neural networks for open domain targeted sentiment. In *Proc. of EMNLP*.

A More Data Statistics

We present the statistics of accumulative percentage of different lengths for targets, opinion spans and offsets in the training data on 4 datasets `14Rest`, `14Rest`, `15Rest` and `16Rest` in Figure 7. As we mentioned in the main paper, similar patterns are observed on accumulative statistics on these 4 datasets. We also present the statistics of the number of targets with a single opinion span and with multiple opinion spans, and the number of opinion associated with a single target span and with multiple target spans, shown in Table 6.



(a) `14Rest`

(b) `14Lap`

(c) `15Rest`

(d) `16Rest`

Figure 7: Accumulative percentage (y-axis) in the training data of different lengths (x-axis) for targets, opinion spans and offsets on the 4 datasets.

B Experimental Details

We test our model on Intel(R) Xeon(R) Gold 6132 CPU, with PyTorch version 1.40. The average run time is 3300 sec/epoch, 1800 sec/epoch, 1170 sec/epoch, 1600 sec/epoch on `14Rest`, `14Rest`, `15Rest` and `16Rest` datasets respectively when $M = 6$. The total number of parameters is 2.5M.

For hyper-parameter, we use pre-trained 300d GloVe (Pennington et al., 2014) to initialize the word embeddings. We use 100 as the embedding size of $\mathbf{w_r}$ (offset embedding). For out-of-vocabulary words as well as $\mathbf{w_r}$, we randomly sample their embeddings from the uniform distribution $\mathcal{U}(-0.1, 0.1)$, as done in (Kim, 2014). We use the bi-directional LSTM with the hidden size 300. We train our model for a maximal of 20 epochs using Adam (Kingma and Ba, 2014) as the optimizer with batch size 1 and dropout rate 0.5 for datasets in restaurant domain and 0.7 for laptop domain. We manually tune the dropout rate from 0.4 to 0.7, and select the best model parameters based on the best F_1 score on the development data and apply it to the test data for evaluation. For experiments with contextualised representation, we adopt the pre-trained language model BERT (Devlin et al., 2019). Specifically, we use bert-as-service (Xiao, 2018) to generate the contextualized word embedding without fine-tuning. We use the representation from the last layer of the uncased version of BERT base model for our experiments.

C Experimental Results

Table 7 presents the experimental result on the previous released dataset by (Peng et al., 2019).

D Decoding based on Viterbi

Let $\mathcal{T} = \{B^{\epsilon}_{j,k}, S^{\epsilon}_{j,k}, I, E, O\}$ as the new tag set under our position-aware tagging scheme, where ϵ denotes the sentiment polarity for the target, and j, k indicate the position information which are the distances between the two ends of an opinion span and the starting position of a target respectively.

As we know, $|j| \leq |k| \leq M, \epsilon \in \{+, 0, -\}$.

$$O(|\mathcal{T}|) = O(|\epsilon|M^2) = O(M^2)$$

We define the sub-tags of $B^{\epsilon}_{j,k}, S^{\epsilon}_{j,k}$ as B and S respectively, and the sub-tags of I, O, E as themselves. We use the bar on top to denote the sub-tag. For example, $\bar{u}$ is the subtag of $u \in \mathcal{T}$.

We use $\pi(i, v)$ to denote the score for the optimal sequence $\{\mathbf{y}_1^* \cdots \mathbf{y}_i^*\}$ among all the possible sequences whose last tag is v.

Given the input $\mathbf{x}$ of length n, we aim to obtain the optimal sequence $\mathbf{y}^* = \{\mathbf{y}_1^* \cdots \mathbf{y}_n^*\}$.

Dataset		# of Target with One Opinion Span	# of Target with Multiple Opinion Spans	# of Opinion with One Target Span	# of Opinion with Multiple Target Spans
`14Rest`	Train	1809	242	1893	193
	Dev	433	67	444	59
	Test	720	128	767	87
`14Lap`	Train	1121	160	1114	154
	Dev	252	44	270	34
	Test	396	67	420	54
`15Rest`	Train	734	128	893	48
	Dev	180	33	224	12
	Test	385	47	438	23
`16Rest`	Train	1029	169	1240	67
	Dev	258	38	304	15
	Test	396	56	452	23

Table 6: Statistics of 4 datasets.

Models	`14Rest`				`14Lap`				`15Rest`				`16Rest`			
	Dev F_1	$P.$	$R.$	F_1	Dev F_1	$P.$	$R.$	F_1	Dev F_1	$P.$	$R.$	F_1	Dev F_1	$P.$	$R.$	F_1
CMLA+	-	40.11	46.63	43.12	-	31.40	34.60	32.90	-	34.40	37.60	35.90	-	43.60	39.80	41.60
RINANTE+	-	31.07	37.63	34.03	-	23.10	17.60	20.00	-	29.40	26.90	28.00	-	27.10	20.50	23.30
Li-unified-R	-	41.44	68.79	51.68	-	42.25	42.78	42.47	-	43.34	50.73	46.69	-	38.19	53.47	44.51
Peng et al. (2019)	-	44.18	62.99	51.89	-	40.40	47.24	43.50	-	40.97	54.68	46.79	-	46.76	62.97	53.62
JET^t $(M = 2)$	47.06	70.00	34.92	46.59	35.00	63.69	23.27	34.08	47.13	64.80	27.91	39.02	42.32	70.76	35.91	47.65
JET^t $(M = 3)$	56.15	73.15	43.62	54.65	43.72	54.18	30.41	38.95	53.23	66.52	33.19	44.28	50.50	66.35	44.95	53.59
JET^t $(M = 4)$	57.47	70.25	49.30	57.94	43.19	57.46	31.43	40.63	58.05	64.77	42.42	51.26	53.57	68.79	48.82	57.11
JET^t $(M = 5)$	59.15	66.20	49.77	56.82	45.47	59.50	33.88	43.17	59.37	64.14	40.88	49.93	54.16	66.86	50.32	57.42
JET^t $(M = 6)$	$\underline{59.51}$	70.39	51.86	**59.72**	$\underline{45.83}$	57.98	36.33	**44.67**	$\underline{60.00}$	61.99	43.74	**51.29**	$\underline{55.88}$	68.99	51.18	**58.77**
JET^o $(M = 2)$	45.02	66.30	35.38	46.14	33.01	50.43	23.88	32.41	46.80	58.88	25.49	35.58	40.33	60.47	39.14	47.52
JET^o $(M = 3)$	53.14	62.31	43.16	50.99	38.99	55.37	33.67	41.88	54.59	55.99	38.02	45.29	47.87	69.45	46.45	55.67
JET^o $(M = 4)$	58.19	63.84	52.44	57.58	40.87	49.86	36.33	42.03	57.14	57.57	42.64	48.99	53.99	73.98	54.41	62.70
JET^o $(M = 5)$	57.94	64.31	54.99	59.29	$\underline{43.23}$	52.36	40.82	**45.87**	59.51	52.02	48.13	50.00	$\underline{56.08}$	66.91	58.71	**62.54**
JET^o $(M = 6)$	$\underline{58.66}$	62.26	56.84	**59.43**	42.50	52.01	39.59	44.96	$\underline{60.32}$	63.25	46.15	**53.37**	55.63	66.58	57.85	61.91
+ Contextualized Word Representation (BERT)																
JET^t $(M = 6)_{\text{+ BERT}}$	61.01	70.20	53.02	60.41	49.07	51.48	42.65	46.65	62.96	62.14	47.25	53.68	60.41	71.12	57.20	63.41
JET^o $(M = 6)_{\text{+ BERT}}$	60.86	67.97	60.32	63.92	45.76	58.47	43.67	50.00	64.12	58.35	51.43	54.67	60.17	64.77	61.29	62.98

Table 7: The experimental results on the previous released datasets ASTE-Data-V1. The underlined scores indicate the best results on the dev set, and the highlighted scores are the corresponding test results.

- Base Case for all the $v \in \mathcal{T}$
  If $v \in \{I, E, O\}$:

  $$\pi(1, v) = \psi_{START,\bar{v}} + f_t(\mathbf{h}_1)_{\bar{\mathbf{v}}}$$

  If $v \in \{B^{\epsilon}_{j,k}, S^{\epsilon}_{j,k}\}$:

  $$\begin{aligned}\pi(1, v) &= \psi_{START,\bar{v}} + \Phi_v(\mathbf{x}, 1) \\ &= \psi_{START,\bar{v}} + f_t(\mathbf{h}_1)_{\bar{\mathbf{v}}} \\ &+ f_s([\mathbf{g}_{1+j,1+k}; \overleftarrow{\mathbf{h}_1}])_{\epsilon} + f_o(\mathbf{g}_{1+j,1+k}) \\ &+ f_r(j, k)\end{aligned}$$

  where $f_t(\mathbf{h}_i)_{\bar{\mathbf{v}}}$, $f_s([\mathbf{g}_{1+j,1+k}; \overleftarrow{\mathbf{h}_1}])_{\epsilon}$, $f_o(\mathbf{g}_{1+j,1+k})$, and $f_r(j, k)$ are the factorized feature score mentioned in the section 2.2.2.

- Loop forward for $i \in \{2, \cdots, n\}$ and all the $v \in \mathcal{T}$
  If $v \in \{I, E, O\}$:

  $$\pi(i, v) = \max_{u \in \mathcal{T}}\{\pi(i-1, u) + \psi_{\bar{u},\bar{v}} + f_t(\mathbf{h}_i)_{\bar{\mathbf{v}}}\}$$

  If $v \in \{B^{\epsilon}_{j,k}, S^{\epsilon}_{j,k}\}$:

  $$\begin{aligned}\pi(i, v) &= \max_{u \in \mathcal{T}}\{\pi(i-1, u) + \psi_{\bar{u},\bar{v}} + \Phi_v(\mathbf{x}, i)\} \\ &= \max_{(u \in \mathcal{T};\ j,k \in [-M,M];\ \epsilon \in \{+,0,-\})}\{ \\ &\quad \pi(i-1, u) + \psi_{\bar{u},\bar{v}} + f_t(\mathbf{h}_i)_{\bar{\mathbf{v}}} \\ &\quad + f_s([\mathbf{g}_{i+j,i+k}; \overleftarrow{\mathbf{h}_i}])_{\epsilon} + f_o(\mathbf{g}_{i+j,i+k}) \\ &\quad + f_r(j, k)\}\end{aligned}$$

- Backtrack for the optimal sequence $\mathbf{y}^* = \{\mathbf{y}^*_1 \cdots \mathbf{y}^*_n\}$

  $$\mathbf{y}^*_n = \arg\max_{v \in \mathcal{T}} \{\pi(n, v) + \psi_{\bar{v},STOP}\}$$

  Loop for $i \in \{n-1, \cdots, 1\}$

  $$\mathbf{y}^*_i = \arg\max_{v \in \mathcal{T}} \{\pi(i, v) + \psi_{\bar{v},\bar{\mathbf{y}}^*_{i+1}}\}$$

Note that $START$ appears before the start of the input sentence and $STOP$ appears after the end of the input sentence.

The time complexity is $O(n|\mathcal{T}|) = O(nM^2)$.

E Analysis

E.1 Robustness Analysis

We present the performance on targets, opinion spans and offsets of different lengths for two models $\textbf{JET}^t(M=6)$ and $\textbf{JET}^o(M=6)$ with BERT on 3 datasets `14Lap`,`15Rest` and `16Rest` in Figure 8, Figure 9 and Figure 10 respectively.

E.2 Qualitative Analysis

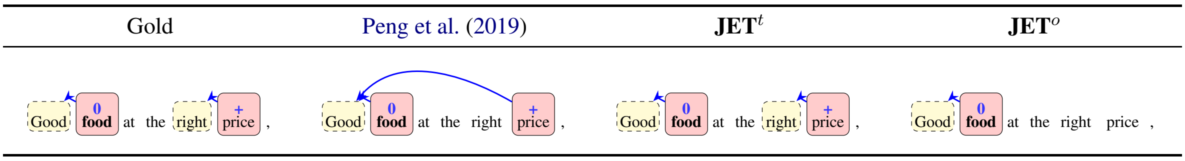
We present one additional example sentence selected from the test data as well as predictions by Peng et al. (2019), $\textbf{JET}^t$ and $\textbf{JET}^o$ in Table 8. As we can see, the gold data contains two triplets. Peng et al. (2019) only predicts 1 opinion span, and therefore incorrectly assigns the opinion span "Good" to the target "price". $\textbf{JET}^t$ is able to make the correct predictions. $\textbf{JET}^o$ only predicts 1 triplet correctly. The qualitative analysis helps us to better understand the differences among these models.

F More Related Work

The task of joint entity and relation extraction is also related to joint triplet extraction. Different from our task, such a relation extraction task aims to extract a pair of entities (instead of a target and an opinion span) and their relation as a triplet in a joint manner. Miwa and Sasaki (2014) and Li and Ji (2014) used approaches motivated by a table-filling method to jointly extract entity pairs as well as their relations. The tree-structured neural networks (Miwa and Bansal, 2016) and CRF-based approaches (Adel and Schütze, 2017) were also adopted to capture rich context information for triplet extraction. Recently, Bekoulis et al. (2018) used adversarial training (Goodfellow et al., 2015) for this task and results show that it performs more robustly in different domains. Although these approaches may not be applied to our task ASTE, they may provide inspirations for future work.

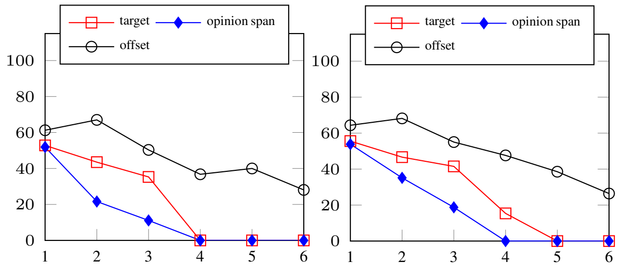
Algorithm 1 Decoding based on Viterbi

```
Initialization for $i = 1$ do
    for $\bar{v} \in \{I, E, O\}$ do
        $v = \bar{v}$
        $\pi(1, v) = \psi_{START,\bar{v}} + f_t(\mathbf{h}_1)_{\bar{\mathbf{v}}}$
    end
    for $\bar{v} \in \{B, S\}$ do
        for $j \in [-M, M]$ do
            for $k \in [j, M]$ do
                for $\epsilon \in \{+, 0, -\}$ do
                    $v = \bar{v}^{\epsilon}_{j,k}$
                    $\pi(1, v) = \psi_{START,\bar{v}} + f_t(\mathbf{h}_1)_{\bar{\mathbf{v}}} + f_s([\mathbf{g}_{1+j,1+k}; \overleftarrow{\mathbf{h}}_1])_{\epsilon} + f_o(\mathbf{g}_{1+j,1+k}) + f_r(j, k)$
                end
            end
        end
    end
end
Loop Forward for $i \in \{2, \cdots, n\}$ do
    for $\bar{v} \in \{I, E, O\}$ do
        $v = \bar{v}$
        $\pi(i, v) = \max_{u \in \mathcal{T}}\{\pi(i-1, u) + \psi_{\bar{u},\bar{v}} + f_t(\mathbf{h}_i)_{\bar{\mathbf{v}}}\}$
    end
    for $\bar{v} \in \{B, S\}$ do
        for $j \in [-M, M]$ do
            for $k \in [j, M]$ do
                for $\epsilon \in \{+, 0, -\}$ do
                    $v = \bar{v}^{\epsilon}_{j,k}$
                    $\pi(i, v) = \max_{u \in \mathcal{T}}\{\pi(i-1, u) + \psi_{\bar{u},\bar{v}} + f_t(\mathbf{h}_i)_{\bar{\mathbf{v}}} + f_s([\mathbf{g}_{i+j,i+k}; \overleftarrow{\mathbf{h}}_i])_{\epsilon} + f_o(\mathbf{g}_{i+j,i+k}) + f_r(j, k)\}$
                end
            end
        end
    end
end
Backward for the optimal sequence $\mathbf{y}^* = \{\mathbf{y}^*_1 \cdots \mathbf{y}^*_n\}$ for $i \in \{n, \cdots, 1\}$ do
    if $i = n$ then
        $\mathbf{y}^*_n = \arg\max_{v \in \mathcal{T}} \{\pi(n, v) + \psi_{\bar{v},STOP}\}$
    end
    else
        $\mathbf{y}^*_i = \arg\max_{v \in \mathcal{T}} \{\pi(i, v) + \psi_{\bar{v},\bar{\mathbf{y}}^*_{i+1}}\}$
    end
end
```



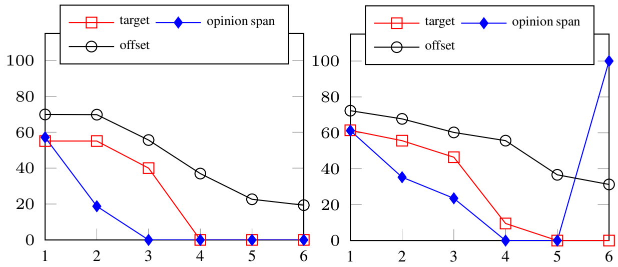
Gold	Peng et al. (2019)	**JET**^{t}	**JET**^{o}
Good 0 **food** at the right + price ,	Good 0 **food** at the right + price ,	Good 0 **food** at the right + price ,	Good 0 **food** at the right price ,

Table 8: Qualitative Analysis



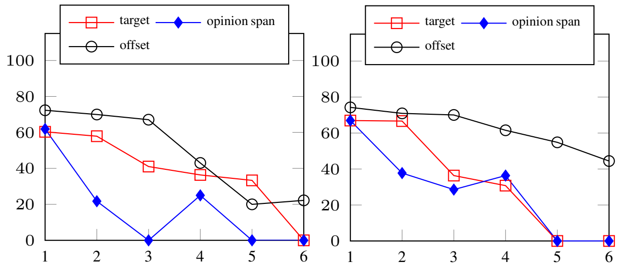
(a) **JET**$^t(M = 6)$ (b) **JET**$^o(M = 6)$

Figure 8: $F_1(\%)$ scores (y-axis) of different lengths (x-axis) for targets, opinion spans and offsets on the dataset `14Lap`.



(a) **JET**$^t(M = 6)$ (b) **JET**$^o(M = 6)$

Figure 9: $F_1(\%)$ scores (y-axis) of different lengths (x-axis) for targets, opinion spans and offsets on the dataset `15Rest`.



(a) **JET**$^t(M = 6)$ (b) **JET**$^o(M = 6)$

Figure 10: $F_1(\%)$ scores (y-axis) of different lengths (x-axis) for targets, opinion spans and offsets on the dataset `16Rest`.