Supplementary Material for 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

1 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 1. 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 1.

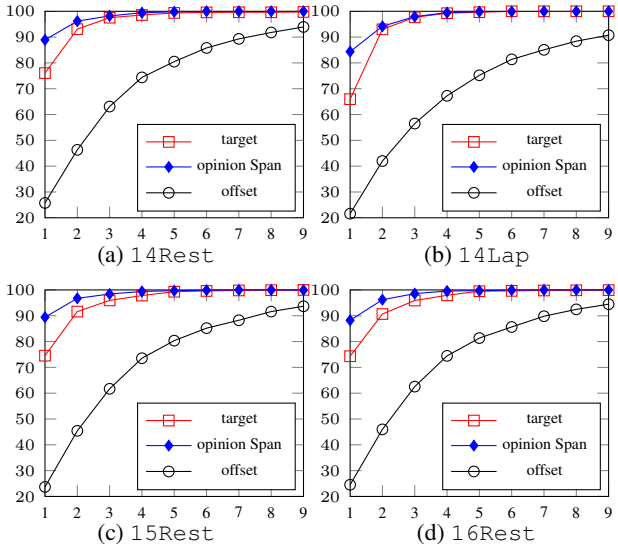


Figure 1: Accumulative percentage (y-axis) in the training data of different lengths (x-axis) for targets, opinion spans and offsets on the 4 datasets.

2 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.

3 Experimental Results

Table 2 presents the experimental result on the previous released dataset by (Peng et al., 2019).

4 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.

∗ 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.

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 1: 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 $(M=6)$	59.51	70.39	51.86	**59.72**	45.83	57.98	36.33	**44.67**	60.00	61.99	43.74	**51.29**	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	43.23	52.36	40.82	**45.87**	59.51	52.02	48.13	50.00	56.08	66.91	58.71	**62.54**
JET^o $(M=6)$	58.66	62.26	56.84	**59.43**	42.50	52.01	39.59	44.96	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 2: 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.

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 $\mathbf{y}^* = \{\mathbf{y}^*_1 \cdots \mathbf{y}^*_n\}$.

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

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

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

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

- 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 $O(n|\mathcal{T}|) = O(nM^2)$.

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

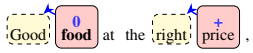
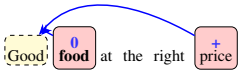
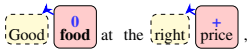
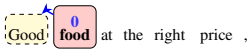
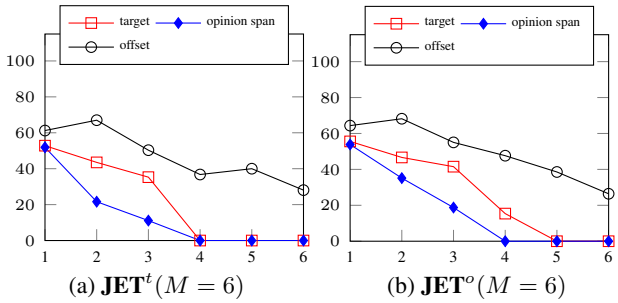
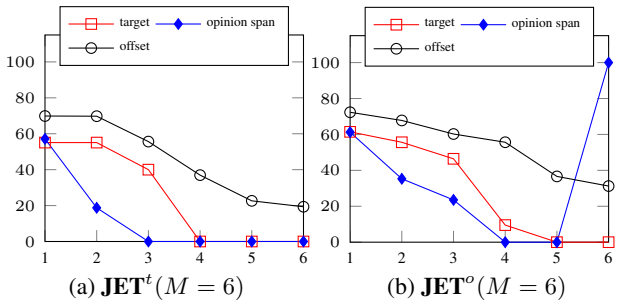
Gold	Peng et al. (2019)	**JET**^{t}	**JET**^{o}
			

Table 3: Qualitative Analysis



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

Figure 2: $F_1(\%)$ 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 3: $F_1(\%)$ scores (y-axis) of different lengths (x-axis) for targets, opinion spans and offsets on the dataset `15Rest`.

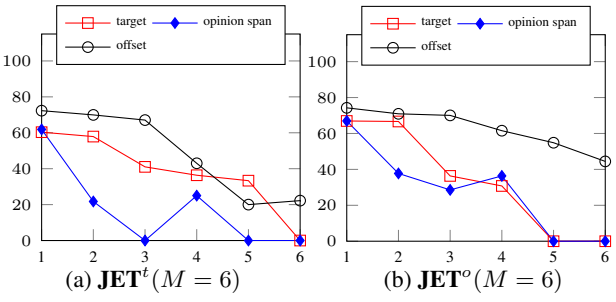
5 Analysis

5.1 Robustness Analysis

We present the performance on targets, opinion spans and offsets of different lengths for two models **JET**$^{t}(M=6)$ and **JET**$^{o}(M=6)$ with BERT on 3 datasets `14Lap`,`15Rest` and `16Rest` in Figure 2, Figure 3 and Figure 4 respectively.

5.2 Qualitative Analysis

We present one additional example sentence selected from the test data as well as predictions by Peng et al. (2019), **JET**^{t} and **JET**^{o} in Table 3. 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". **JET**^{t} is able to make the correct predictions. **JET**^{o} only predicts 1 triplet correctly. The qualitative analysis helps us to better understand the differences among these models.



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

Figure 4: $F_1(\%)$ scores (y-axis) of different lengths (x-axis) for targets, opinion spans and offsets on the dataset `16Rest`.

6 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.

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*.

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*.

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

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*.

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

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*.

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*.

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

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