Boundary-Driven Table-Filling for Aspect Sentiment Triplet Extraction

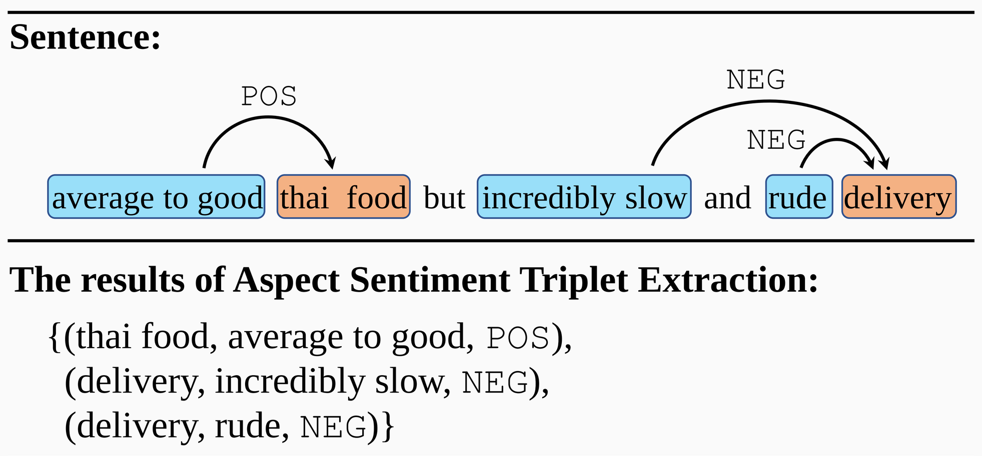
EMNLP 2022

Yice Zhang; Yifan Yang; Yihui Li; Bin Liang;
Shiwei Chen; Yixue Dang; Ming Yang; Ruifeng Xu
Harbin Insitute of Technology, Shenzhen, China

The 2022 Conference on Empirical Methods in Natural Language Processing
7-11 Dec '2022 - Abu Dhabi (The United Arab Emirates)

Introduction

Aspect Sentiment Triplet Extraction (ASTE) aims to extract the aspect terms along with the corresponding opinion terms and the expressed sentiments in the review, exemplified in the right figure.

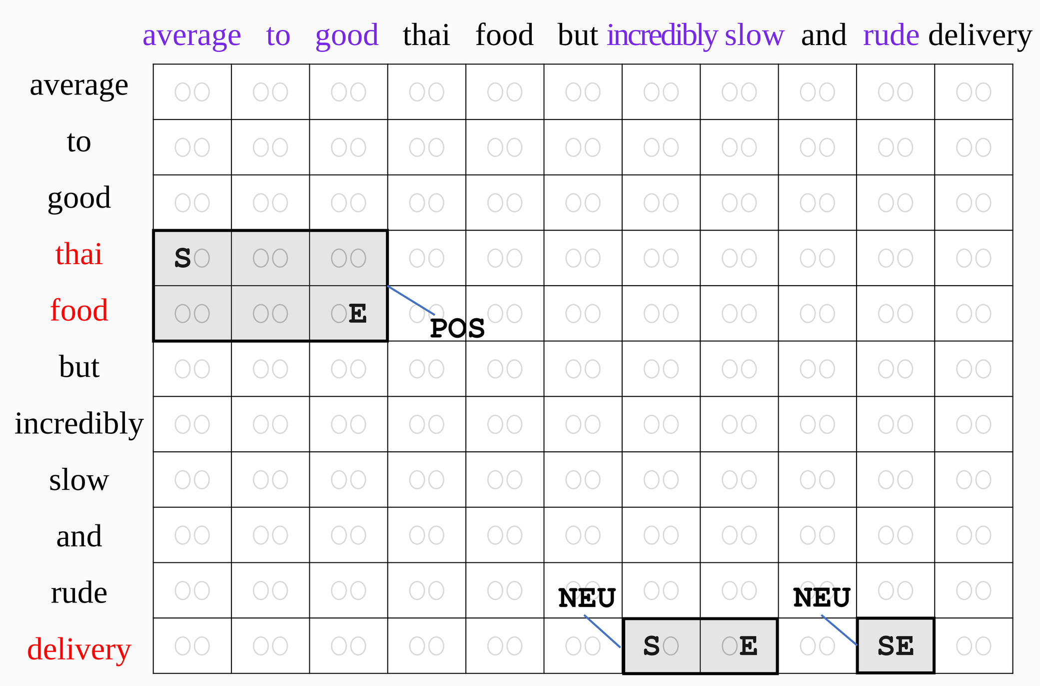


Previous methods tackle the ASTE task through a table-filling approach, where the triplets are represented by a two-dimensional (2D) table of word-pair relations. This formalization suffers from **relation inconsistency** and **boundary insensitivity** when dealing with multi-word aspect terms and opinion terms. Researchers attempt to solve the issue of relation inconsistency through a span-based method, but their method discards fine-grained word-level information, which is the advantage of the table-filling approach.

This paper proposes a Boundary-Driven Table-Filling (BDTF) approach for ASTE to overcome the above limitations. BDTF represents each triplet as a relation region in the 2D table and transforms the ASTE task into the detection and classification of relation regions. Classification over the entire relation region ensures relation consistency, and those relation regions with boundary errors can be removed by being classified as invalid. We also notice that the quality of the table representation greatly affects the performance of BDTF. Therefore, we also develop an effective relation representation learning approach to learn the table representation.

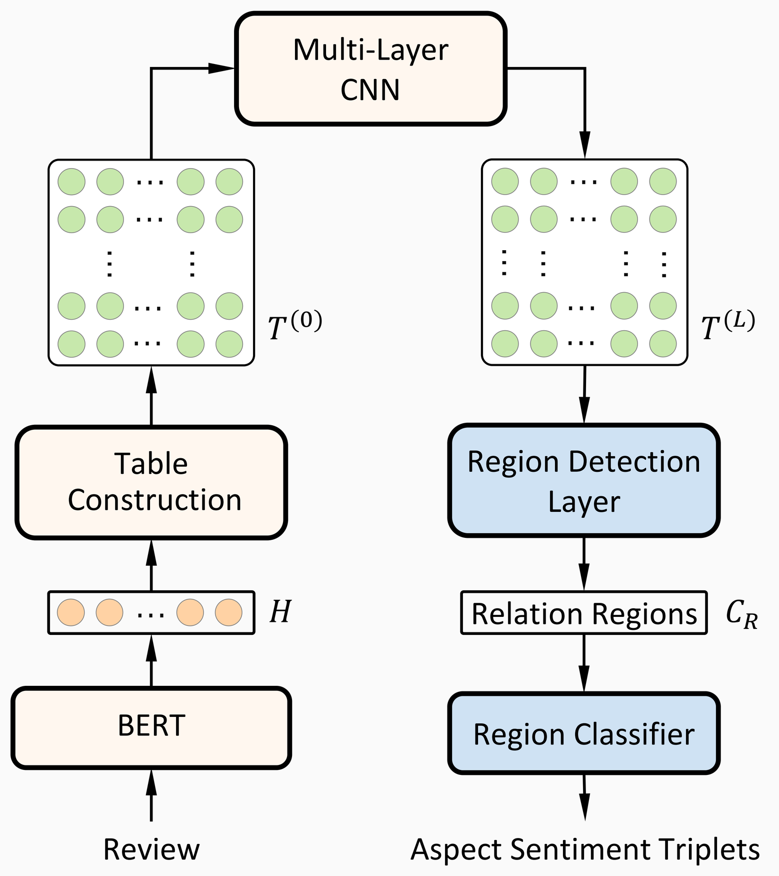
Task Formalization

Given a sentence X of length n, the goal of ASTE is to extract a set of aspect sentiment triplets. A triplet is defined as (*aspect*, *opinion*, *sentiment*) where *sentiment* $\in$ {POS, NEU, NEG}. We represent a triplet as a relation region in the 2D table. Its boundary indicates the position of *aspect* and *opinion*, and its type indicates *sentiment*. Relation regions are located by two boundary tags. S denotes the upper left corner, and E denotes the lower right corner.



Our Model

We briefly present the proposed approach in the right figure. Our model can be divided into Representation Learning and Extraction Module.



Representation Learning. We develop a relation representation learning approach to learn the table representation. This consists of three parts:

1. We first learn the word-level contextualized representations of the input review through BERT.
2. Then, we adopt a tensor-based operation to construct the relation-level representations to fully exploit the word-to-word interactions.
3. Finally, we model relation-to-relation interactions through a multi-layer convolution-based encoder to enhance the relation-level representations.

$$
\begin{aligned}
h_1, h_2, \cdots, h_T &= \mathrm{BERT}(x_1, x_2, \cdots, x_T).\\
c_{ij} &= \mathrm{pooling}(h_i, h_{i+1}, \cdots, h_j),\\
t_{ij} &= \mathcal{T}(h_i, h_j; V),\\
r_{ij}^{(0)} &= f(\mathrm{Linear}([h_i; h_j; c_{ij}; t_{ij}])).
\end{aligned}
\qquad
\begin{aligned}
T^{(0)} &= \begin{pmatrix} r_{11}^{(0)} & \cdots & r_{1n}^{(0)} \\ \vdots & \ddots & \vdots \\ r_{n1}^{(0)} & \cdots & r_{nn}^{(0)} \end{pmatrix},\\
T^{(l+1)} &= \mathrm{CNN}(T^{(l)}).
\end{aligned}
$$

Extraction Module. For each element in the 2D table, the region detection layer utilizes two classifiers to calculate the probability of its boundary tag being S and E. Then, we select the top-k S candidates and top-k E candidates by P_{ij}^{S} and P_{ij}^{E}, respectively. These valid S-E pairs of the selected candidates form the region candidate pool.

$$
\begin{aligned}
P_{ij}^{\mathrm{S}} &= \mathrm{sigmoid}\left(\mathrm{Linear}\left(r_{ij}^{(L)}\right)\right),\\
P_{ij}^{\mathrm{E}} &= \mathrm{sigmoid}\left(\mathrm{Linear}\left(r_{ij}^{(L)}\right)\right).\\
k &= \max(n \cdot z, k_{min}),
\end{aligned}
\qquad
\begin{aligned}
p_{abcd}^{(L)} &= \mathrm{pooling}\begin{pmatrix} r_{ab}^{(L)} & \cdots & r_{ad}^{(L)} \\ \vdots & \ddots & \vdots \\ r_{cb}^{(L)} & \cdots & r_{cd}^{(L)} \end{pmatrix},\\
r_{abcd} &= \left[r_{ab}^{(L)}; r_{cd}^{(L)}; p_{abcd}^{(L)}\right].\\
P_{abcd}(y_{\mathcal{T}}) &= \mathrm{softmax}(\mathrm{Linear}(r_{abcd})).
\end{aligned}
$$

Given a relation region determined by S(a, b) and E(c, d), we concatenate the S representation, the E representation, and the max-pooling result of the relation matrix over this region as its feature representation. Then we use a classifier to predict its type $y_{\mathcal{T}} \in$ {POS, NEU, NEG, Invalid}. At inference, we drop those relation regions whose predicted types are Invalid and generate the aspect sentiment triplets from the remaining relation regions.

Result and Anlysis

Main Result. According to these results on four ASTE datasets, our approach consistently attains the best performance, demonstrating its effectiveness. Although EMC-GCN and SSJE introduce the additional word dependency information in representation learning through Graph Convolutional Network (GCN), our approach still outperforms them.

Model	Rest 14	Lap 14	Rest 15	Rest 16
Two-stage(Peng et al., 2020)	51.46	42.87	52.32	54.21
JET$^o_{M=6}$(BERT)(Xu et al., 2020)	62.40	51.04	57.53	63.83
BMRC(Chen et al., 2021a)	68.64	58.18	58.79	67.35
Table-Filling Approaches				
OTE-MTL♮(Zhang et al., 2020)	59.71	44.78	47.94	56.82
GTS-BERT♮(Wu et al., 2020)	67.50	54.36	60.15	67.93
Double-Encoder(Jing et al., 2021)	69.55	59.11	59.27	70.44
EMC-GCN(Chen et al., 2022)	71.78	58.81	61.93	68.33
Span-Based Approaches				
Span-ASTE(Xu et al., 2021)	71.85	59.38	63.27	70.26
SSJE(Li et al., 2022a)	72.26	60.41	65.05	71.38
SSJE w/o GCN(Li et al., 2022a)	71.33	59.48	62.55	69.81
BDTF (Ours)	**74.35**	**61.74**	**66.12**	**72.27**

Comparison of Table-Filling Approaches. We replace our extraction module with TF_{GTS} and TF_{Double} while keeping the rest unchanged. Results show that our table-filling approach significantly outperforms the previous approaches on all four datasets.

Decoding	Rest 14	Lap 14	Rest 15	Rest 16	AVG-Δ
TF_{GTS}(Wu et al., 2020)	70.00	56.60	59.58	68.75	-
TF_{Double}(Jing et al., 2021)	69.04	54.98	59.75	67.61	-
BDTF(Ours)	**74.35**	**61.74**	**66.12**	**72.27**	**+4.85**

Besides, compared with the previous table-filling approaches, our approach significantly reduces the proportion of boundary errors ($B.$) and multi-word relation errors ($R_M.$) in the predictions.

Model	Rest 14		Lap 14		Rest 15		Rest 16	
	$B.$	R_M	$B.$	R_M	$B.$	R_M	$B.$	R_M
TF_{GTS}	10.16	6.02	11.75	7.97	14.37	6.02	9.21	4.42
BDTF	6.77	3.33	6.03	6.70	10.50	4.16	5.34	4.35
Δ	-3.39	-2.69	-5.75	-1.27	-3.87	-1.86	-3.87	-0.07

Relation Learning Analysis. Our model performs poorly when the relation representation is obtained only by feature concatenation. Adding context, tensor-based operation, or CNN results in significant performance improvements. Adding these three components together improves the average performance by 8.71%. We also compare some element-wise operations: addition, subtraction, and multiplication. However, their performance improvements are not significant enough and sometimes even negative.

Model	Rest 14	Lap 14	Rest 15	Rest 16	AVG-Δ
Concat	68.13	52.31	55.11	64.09	-
Concat + Context	72.96	59.25	64.96	72.12	+7.41
Concat + Tensor	71.63	58.78	63.77	70.17	+6.18
Concat + CNN	71.91	55.01	64.81	69.39	+5.37
Full Model	**74.35**	**61.74**	**66.12**	**72.27**	**+8.71**
Concat + Add	67.78	53.43	57.78	65.63	+1.24
Concat + Sub	69.56	52.49	55.97	66.27	+1.16
Concat + Mul	69.80	53.34	58.08	66.76	+2.08