

Nonautoregressive Encoder–Decoder Neural Framework for End-to-End Aspect-Based Sentiment Triplet Extraction

Hao Fei^{ID}, Yafeng Ren, Yue Zhang^{ID}, *Member, IEEE*, and Donghong Ji

Abstract—Aspect-based sentiment triplet extraction (ASTE) aims at recognizing the joint triplets from texts, i.e., aspect terms, opinion expressions, and correlated sentiment polarities. As a newly proposed task, ASTE depicts the complete sentiment picture from different perspectives to better facilitate real-world applications. Unfortunately, several major challenges, such as the overlapping issue and long-distance dependency, have not been addressed effectively by the existing ASTE methods, which limits the performance of the task. In this article, we present an innovative encoder–decoder framework for end-to-end ASTE. Specifically, the ASTE task is first modeled as an unordered triplet set prediction problem, which is satisfied with a nonautoregressive decoding paradigm with a pointer network. Second, a novel high-order aggregation mechanism is proposed for fully integrating the underlying interactions between the overlapping structure of aspect and opinion terms. Third, a bipartite matching loss is introduced for facilitating the training of our nonautoregressive system. Experimental results on benchmark datasets show that our proposed framework significantly outperforms the state-of-the-art methods. Further analysis demonstrates the advantages of the proposed framework in handling the overlapping issue, relieving long-distance dependency and decoding efficiency.

Index Terms—Bipartite matching loss, encoder–decoder framework, natural language processing (NLP), nonautoregressive decoding, pointer network, sentiment analysis.

I. INTRODUCTION

ASPECT-BASED sentiment analysis (ABSA) aims to mine the sentiment and opinion of targeted aspects behind the texts. It has received much research attention in the community of data mining and knowledge discovering [1]–[7].

Manuscript received May 13, 2021; revised August 21, 2021; accepted November 15, 2021. This work was supported in part by the National Natural Science Foundation of China under Grant 61772378 and Grant 62176187, in part by the National Key Research and Development Program of China under Grant 2017YFC1200500, in part by the Research Foundation of Ministry of Education of China under Grant 18JZD015, in part by the Science of Technology Project of Guangzhou under Grant 20210202607, and in part by the Key Project of State Language Commission of China under Grant ZDI135-112. (*Corresponding author: Donghong Ji.*)

Hao Fei and Donghong Ji are with the School of Cyber Science and Engineering, Wuhan University, Wuhan 430072, China (e-mail: hao.feifei@whu.edu.cn; dhji@whu.edu.cn).

Yafeng Ren is with the Laboratory of Language and Artificial Intelligence, Guangdong University of Foreign Studies, Guangzhou 510420, China (e-mail: renyafeng@whu.edu.cn).

Yue Zhang is with the School of Engineering, Westlake University, Hangzhou 310024, China (e-mail: yue.zhang@wias.org.cn).

Color versions of one or more figures in this article are available at <https://doi.org/10.1109/TNNLS.2021.3129483>.

Digital Object Identifier 10.1109/TNNLS.2021.3129483

- (a) The hot dogs and the smoked salmon are fabulous, as well as the service here.
Opinion triplets: <The hot dogs, fabulous, positive>
 <the smoked salmon, fabulous, positive>
 <the service, fabulous, positive>
- (b) Most fabulous service, foods (both the hot dogs and smoked salmon) ever tried.
Opinion triplets: <service, Most fabulous, positive>
 <foods, Most fabulous, positive>
 <the hot dogs, Most fabulous, positive>
 <smoked salmon, Most fabulous, positive>
- (c) Service is nice, as well as the ambience and the takeout, except for the dessert, and the cheeseburgers, too steep price!
Opinion triplets: <Service, nice, positive>
 <the ambience, nice, positive>
 <the takeout, nice, positive>

 <the dessert, too steep price, negative>
 <the cheeseburgers, too steep price, negative>

Fig. 1. Examples of aspect-based sentiment triplet ((aspect term, opinion expression, and sentiment polarity)) extraction.

ABSA includes several closely related subtasks, such as aspect term extraction (ATE), opinion terms extraction (OTE), and aspect sentiment classification (ASC), which together depict the complete sentiment from different aspects, where ATE indicates what the targeted aspects are, ASC shows how their sentiment polarities are, and OTE emphasizes why have such polarities [8]. However, previous work focuses on handling the above subtasks individually or jointly with two subtasks [1], [9]–[14] but does not exploit the triadic relations among each other. Peng *et al.* [8] pioneered the unified task, finding all joint triplets (aspect terms, opinion terms, and the sentiment polarities) of a given text, namely, aspect-based sentiment triplet extraction (ASTE), as exemplified in Fig. 1. Unfortunately, they adopt a two-stage framework and fail to obtain competitive performance since such pipeline method is much prone to error propagation [15]. More recently, some efforts try to perform end-to-end ASTE with joint models using sequence labeling [16], [17] and table-filling method [18]. However, existing studies leave several important issues unaddressed, which limits further task improvements.

First, there is a large number (around 38% according to experiments) of the overlaps between aspect and opinion terms in different triplets. Modeling of such overlapping cases can be highly important to task performance. However, the existing joint models solve the issue by sequential labeling with sophisticatedly enhanced “BIO” tagsets, which can cause large costs on label searching [19], [20]. Second, previous

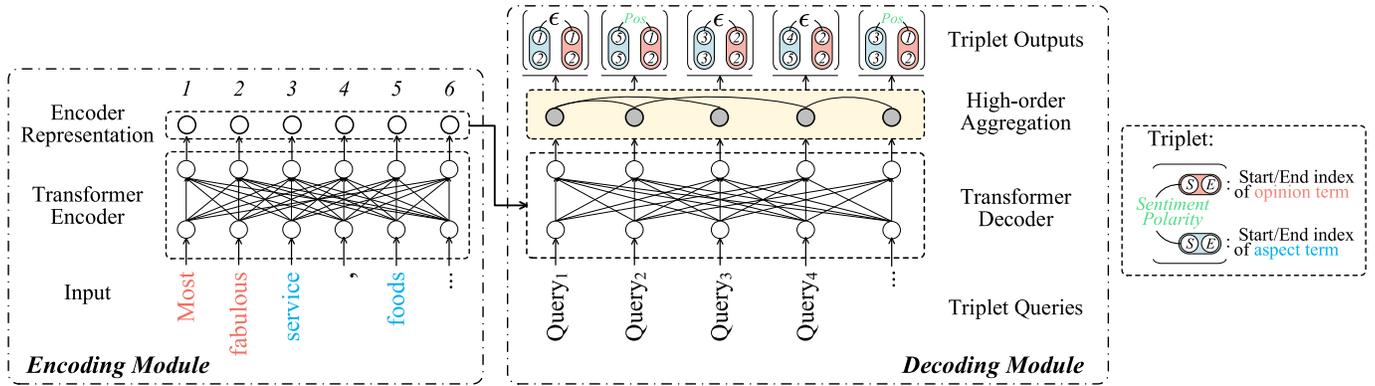


Fig. 2. Overall framework of the proposed method. The encoding module yields contextual representations, which are used for decoding module. The decoder takes as input fixed length of triplet queries and performs nonautoregressive parallel decoding based on pointer network, during which the high-order aggregation is performed. Finally, the system produces all possible sentiment triplets in one shot.

studies largely suffer from the long-range dependency problem due to the sequential extraction model architecture [21]–[23]. While the table-filling method in [18] may handle the above two issues to some extent, it has the cost of high model complexity [i.e., $O(n^2)$] and thus lower decoding efficiency. Third, overlapping structures share rich mutual interactions that may help to better induce sentiment polarities (e.g., with better polarity consistency), which have not been exploited effectively by the existing methods. For example, different aspects under the same opinion expression may have the same sentiment polarity, but the same aspect with different opinion expressions can hold distinct sentiment tendencies.

We aim to address the above issues by using a novel encoder–decoder framework [24], modeling the triplet extraction as a structure prediction problem. As shown in Fig. 2, we use the transformer encoder [25] to obtain contextual representations. In the decoder side, all possible triplets will be produced jointly, where the start/end boundaries of aspect terms and opinion terms are generated in parallel by a pointer network (also known as pointer net), and the corresponding sentiment polarities are decided. In particular, all the preidentified aspect–opinion pairs are grouped into clusters according to their opinion-aware connectivity, which are then encoded and aggregated by our high-order aggregation module (cf. Fig. 3) for sentiment assignment. With this, our system can fully explore the underlying interactions between the aspect–opinion terms within the overlapping structures.

We observe that the outputs in the ASTE task have no intrinsic order in nature. Taking the sentences (a) and (b) of Fig. 1 as examples, although the aspect and opinion terms appear in a different order in the text, the final opinion triplets are kept (almost) the same and unordered. On the other hand, in the autoregressive encoder–decoder architecture [24], the permutation-sensitive cross-entropy loss imposes penalties for the predicted triples out of positions. To combat that, we first reformulate ASTE as an unordered triplet set prediction problem, where specifically, the pointer net performs decoding under nonautoregressive parallel scheme (cf. Fig. 2). We then propose a bipartite matching loss (cf. Fig. 4) for the learning of our nonautoregressive system, which is invariant to the permutation of predictions and thus more capable of

evaluating the differences between the gold triplets and the predicted triplets.

Extensive experiments on four benchmark datasets show that our framework significantly outperforms the previous state-of-the-art methods, demonstrating that the proposed nonautoregressive decoding and matching loss are highly beneficial for the ASTE task. Further in-depth analysis is conducted for revealing the advantages of our method in different aspects. We show that our system can better handle the overlapping issue and long-distance dependency problem while maintaining stronger robustness and higher decoding efficiency, compared with previous baseline methods.

We summarize our contributions in this work as follows.

- 1) We cast the ASTE task as a triplet set prediction problem and handle it with a pointer-net-based encoder–decoder framework. To our knowledge, we are the first presenting an encoder–decoder-based model with pointer network for end-to-end ASTE.
- 2) We propose a novel opinion-aware high-order aggregation module for better sentiment classification (SC), fully exploring the underlying interactions between the aspect–opinion terms within the overlapping structure.
- 3) We model ASTE as an unordered triplet set prediction and propose to perform nonautoregressive decoding with the transformer-based pointer net. We further introduce a bipartite matching loss for training the nonautoregressive system.
- 4) Our system achieves state-of-the-art performances on multiple benchmark datasets. Experimental analysis shows that our system can better handle the overlapping issue and long-distance dependency problem while maintaining stronger robustness and higher decoding efficiency compared with the previous methods.

II. RELATED WORK

A. Sentiment Analysis

Sentiment analysis and opinion mining, inferring the sentiment intensity given a text, as one of the hottest research topics in the natural language processing (NLP) and data mining community, has long gained much attention [1]–[3],

[7], [26]. Later, the research focus has been shifted into the ABSA (i.e., fine-grained sentiment analysis), determining the sentiment polarities toward the specific aspects in the sentence [27], [28]. Compared with the standard coarse-grained level sentiment analysis, such a fine-grained analysis shows more impacts on the real-world texts, such as social media texts and product reviews, and thus can facilitate a wide range of downstream applications [29]–[31]. Within past decades, a large volume of sentiment analysis work has been proposed [32]–[38]. Prior methods for sentiment analysis (either at coarse- or fine-grained level) mostly employ statistical machine learning models with discrete features that are manually crafted [39]–[42]. Later, neural networks together with continuous distributed features are extensively adopted for enhancing the task performances for sentiment analysis [32], [33], [43], [44].

Generally, there are three subtasks that closely relate to ABSA, including the ATE, the OTE, and the aspect term sentiment classification (ATC). Each of the subtasks answers the question of fine-grained sentiment analysis in three different perspectives. For example, ATE indicates “WHAT” the targeted aspects are, ATC shows “HOW” their sentiment polarities are, and OTE emphasizes “WHY” have such polarities [8], [45]. Currently, there are lines of efforts that are extensively paid for those sentiment-based subtasks individually or in a combination of any two [1], [9]–[14]. In the initial works, some memory-based models are proposed for SC based on the preidentified aspects or targets [32], [46]. Later, Toh and Wang [47] started the ATE task via a conditional random field (CRF) model. Follow-up works are presented to improve the ATE task by proposing more sophisticated neural models [48]–[50]. Another line of effort is paid for the extraction of opinion terms, mining the underlying causes of the sentiments [51]–[54]. Later, the aspect and opinion term coextraction task has been explored, which leverages the mutual benefits of these two subtasks, i.e., detecting the sentiment targets along with the corresponding reasons [1], [9], [12], [55].

B. Aspect-Based Sentiment Triplet Extraction

In fact, these fine-grained subtasks (i.e., ATE, OTE, and ATC) exploiting the triadic relations among each other can be mutually complementary to each other. Therefore, Peng *et al.* [8] unified these tasks into one complete solution, i.e., ASTE. They show that by solving the one unified task, the subtasks can be further improved. Intuitively, the ASTE task explores the maximum mutual benefits of the above sentiment subtasks. Unfortunately, Peng *et al.* [8] adopted a pipeline method for the ASTE. In their framework, they perform aspect terms with sentiment labeling extraction; the OTE is in the first stage, and in the second stage, the aspect terms will be coupled with the opinion terms, forming the final triplets. We note that such a pipeline scheme may suffer from error propagation, i.e., the semantic interactions between the two stages are not explicitly modeled [15].

Later, several joint methods are presented for better ASTE performances. Chen *et al.* [16] introduced a hierarchical

sequence labeling model to tag the aspect terms, opinion expressions, and the sentiment polarity. Xu *et al.* [17] constructed a position-aware tagging approach to extract the triplets by encoding the interactions among the elements within a triplet, which can help to improve the ASTE task. Then, a table-filing framework is proposed by Wu *et al.* [18] for searching out the aspect terms and opinion expressions together with the correlated polarities in the rows and columns of the table. Very recently, Chen *et al.* [56] transformed the triplet prediction into a machine reading comprehension task and solve it with an end-to-end manner and become the current state-of-the-art model. However, as we pointed out earlier, these works could be still incompetent on handling some challenges in ASTE, such as triplet overlap issue, long-range dependency issue, and modeling the mutual interactions between the substructure of triplets, which hinders the further ASTE task improvements. In this work, we aim to solve these problems by modeling the task as an unordered triplet set prediction and presenting a nonautoregressive framework based on the pointer network.

C. Encoder–Decoder Framework

This work closely relates to the application of encoder–decoder framework (also known as sequence-to-sequence architecture) of neural models [24]. In the encoder–decoder structure, the length of output sequence is not restricted to be the same of the input sequence length such that the scheme can be widely employed for the asynchronous sequence generation tasks in NLP, such as neural machine translation [57]–[59], text summarization [60]–[62], dialogue system [63]–[65], and structure prediction [66]–[68]. In this work, we adopt the pointer network [69] as our backbone model for generating the start/end position of the aspect and opinion terms. A pointer network is built based on the encoder–decoder paradigm, functioning by learning the conditional probability of an output at the positions corresponding to the input tokens. The pointer network is advanced in making decisions by consulting all the input elements in a global scope [66], [70], [71].

In general encoder–decoder architecture, the decoding process mostly follows the ordered autoregressive sequence [24], i.e., together with the cross-entropy loss. However, we further observe that the triplets have no intrinsic textual order in the ASTE task. We thus propose to perform nonautoregressive decoding [72]–[74], with the transformer-based pointer net, outputting all the boundary positions in parallel. We also introduce a bipartite matching loss for better facilitating the model training.

III. PRELIMINARY

A. Task Formulation

The goal of ASTE is to detect all possible triples in a given text, which is modeled as a ⟨aspect-opinion-polarity⟩ set prediction problem. Technically, given an input sentence $S = \{w_1, \dots, w_T\}$, a system is expected to output a set of triplets $Y = \{\dots, y_i, \dots\}$ and $y_i = \langle a_m, o_n, c \rangle \in A \times O \times C$, where $A = \{a_1, \dots, a_M\}$ are all possible aspect terms, $O = \{o_1, \dots, o_N\}$ are the associated opinion expressions,

and $C = \{\text{Pos}, \text{Neg}, \text{Neu}, \epsilon\}^1$ are the corresponding sentiment polarity labels, with a dummy label ϵ indicating the invalid relation between the aspect and the opinion term. Taking the first sentence (a) in Fig. 1 as an example, the system should output three triplets: “(The hot dogs, fabulous, positive),” “(the smoked salmon, fabulous, positive),” and “(the service, fabulous, positive).”

B. Encoder–Decoder With Pointer Network

In the encoder–decoder framework, the decoder equipped with pointer [69] picks elements from input tokens with the highest probability at each decoding frame. The pointer employs attention mechanism [57], [58] to select targets from the input sequence. Technically, given the encoder representations $\mathbf{H} = [\mathbf{h}_1, \dots, \mathbf{h}_T]$ of input tokens and the current decoding representation s_i , we calculate and normalize the relatedness score between s_i and each \mathbf{h}_j

$$\begin{aligned} v_{ij} &= \text{Score}(s_i, \mathbf{h}_j) \\ &= \text{Tanh}(s_i^T \mathbf{W}_1 \mathbf{h}_j + \mathbf{U}_1^T s_i + \mathbf{U}_2^T \mathbf{h}_j + b) \\ o_{ij} &= \text{Softmax}(v_{ij}), \quad j = [1, \dots, T] \end{aligned} \quad (1)$$

where \mathbf{W}_1 , \mathbf{U} , and b are parameters. We then take the position j^* with the maximal relatedness probability o_{ij^*} as the output of the i th decoding step, formalized as

$$P_i = j^* = \underset{1 \leq j \leq T}{\text{Argmax}}(o_{i1}, \dots, o_{iT}) \quad (2)$$

where P_i denotes the position that the current pointer directs to. Note that since each pointer decision is made by consulting all input tokens, the model can utilize the global information. We summarize the pointer procedure as follows:

$$P_i = \text{Ptr}(s_i | \mathbf{H}). \quad (3)$$

IV. FRAMEWORK

As shown in Fig. 2, our proposed framework has two main components: the encoding module and the decoding module. We employ transformer (Trm) [25] to produce contextual representations of the input tokens. A transformer decoder is then used to detect the start and end boundaries of the terms of the aspect–opinion pairs based on the embeddings of triplet queries. Finally, the high-order aggregation layer further determines the sentiment polarities of each aspect–opinion pair.

A. Encoding

1) *Input Representation*: The input representations are derived from three sources. We first construct the vector representation \mathbf{x}_t^w of each word w_t from pretrained embeddings [75]. We additionally represent the absolute position for each word as an embedding \mathbf{x}_t^p for enhancement. Moreover, a convolutional neural network (CNN) is used to encode the characters inside each word into a character-level word

¹Pos, Neg, and Neu are the abbreviation of three labels of sentiment polarity, Positive, Negative, and Neutral.

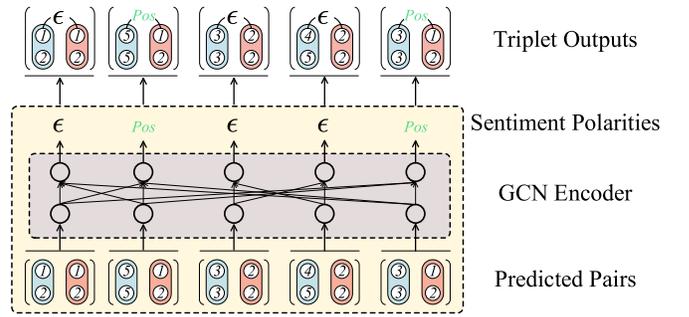


Fig. 3. High-order aggregation layer for sentiment polarity detection.

representation \mathbf{x}_t^c . Finally, the total input representation is the concatenation of all above elements

$$\mathbf{x}_t = [\mathbf{x}_t^w; \mathbf{x}_t^p; \mathbf{x}_t^c] \quad (4)$$

where $[\cdot]$ refers to the concatenation operation.

2) *Contextual Encoder*: Transformer has been shown prominent on learning the interaction between each pair of input words, leading to better contextualized word representations [25]. We here use a multilayer transformer encoder. Technically, transformer attention computes the relatedness between \mathbf{K} and \mathbf{Q} via self-attention mechanism

$$\alpha = \text{Softmax}\left(\frac{\mathbf{Q} \cdot \mathbf{K}^T}{\sqrt{d_k}}\right) \cdot \mathbf{V} \quad (5)$$

where d_k is a scaling factor, and the queries \mathbf{Q} , values \mathbf{V} , and keys \mathbf{K} are the same of input representations \mathbf{x} . All the vectors produced by m parallel attention heads are concatenated together to form a unified representation

$$\mathbf{H} = [\alpha_1; \dots; \alpha_m] \cdot \mathbf{W}_2 + b_\alpha. \quad (6)$$

We summarize all the encoding step as follows:

$$\mathbf{H} = \{\mathbf{h}_1, \dots, \mathbf{h}_T\} = \text{Trm}^{\text{enc}}(\mathbf{x}_1, \dots, \mathbf{x}_T) \quad (7)$$

where \mathbf{h}_i is the desired contextual encoding representations.

B. Decoding

Each step at the corresponding decoder frame outputs the corresponding decoding representation s_i . We treat the ASTE task as an unordered set prediction problem, thus requiring the decoder to yield all the output triplets in a parallel manner. Also, the transformer becomes our first choice as it is able to perform nonautoregressive decoding. The transformer decoder can also retrieve features from bidirectional contexts with maximum information integration.

1) *Decoding Input*: Our nonautoregressive transformer decoder generates all the outputs in one shot by taking all the decoding input. Instead of using the representations from the encoder side, we maintain a set of triplet queries in a fixed size D as decoding inputs. The size D refers to the maximum volume of the triplets that we expect the system to generate. We set D significantly larger than the potential number of triplets in the sentence in advance. We transform the triplet queries $\{q_1, \dots, q_D\}$ into trainable vectorial embeddings $\{e_1^q, \dots, e_D^q\}$.

To further enhance the capability of the pointer, we add another set of position embeddings $\{e_1^p, \dots, e_D^p\}$ for each decoding step, induced the same way as for encoding part. We concatenate the triplet query embedding and the position embedding as the unified decoding input

$$\mathbf{e}_i = [e_i^p; e_i^q]. \quad (8)$$

2) *Nonautoregressive Pointer Decoder*: We use an n -layer stacked transformer as the nonautoregressive decoder

$$\{s_1, \dots, s_D\} = \text{Trm}^{\text{dec}}(e_1, \dots, e_D) \quad (9)$$

where s_i is the decoding representation. Each frame (i.e., the pointer) in transformer decoder independently derives a pair of aspect (A) and opinion (O) terms, where each term involves the start (s) and end (e) index to correspond to the position of certain input token w_i . Specifically, there are total four independent positions [i.e., $P^{(A,s)}$, $P^{(A,e)}$, $P^{(O,s)}$ and $P^{(O,e)}$] that the pointer should direct to.

We first perform a nonlinear transformation on decoding representation s_i to obtain independent position feature representations by using four separate feedforward networks (FFNs)

$$\begin{aligned} s_i^{A,s} &= \text{FFN}(s_i) \\ s_i^{A,e} &= \text{FFN}(s_i) \\ s_i^{O,s} &= \text{FFN}(s_i) \\ s_i^{O,e} &= \text{FFN}(s_i). \end{aligned} \quad (10)$$

We then put the pointer calculation [via (3)] on these position feature representations to output these positions

$$\begin{aligned} P_i^{A,s} &= \text{Pointer}(s_i^{A,s} | \mathbf{H}) \\ P_i^{A,e} &= \text{Pointer}(s_i^{A,e} | \mathbf{H}) \\ P_i^{O,s} &= \text{Pointer}(s_i^{O,s} | \mathbf{H}) \\ P_i^{O,e} &= \text{Pointer}(s_i^{O,e} | \mathbf{H}). \end{aligned} \quad (11)$$

We resolve the aspect and the opinion terms via their position indices and further construct the representations of these terms

$$\mathbf{r}_i^{A/O} = [\mathbf{h}_{P_i^{A/O,s}}; \mathbf{h}_{P_i^{A/O,e}}; \mathbf{h}^{A/O,*}; \mathbf{h}^{A/O,l}] \quad (12)$$

where $\mathbf{h}_{P_i^{(A/O,s/e)}}$ is the boundary representation of the terms, $\mathbf{h}^{A/O,l}$ is the embedding vector for the span width, and $\mathbf{h}^{A/O,*}$ is the span attention representation over the term tokens

$$\begin{aligned} \mathbf{v}_t &= \mathbf{V} \cdot \tanh(\mathbf{W}_3 \cdot \mathbf{h}_t) \\ \alpha_t &= \text{Softmax}(\mathbf{v}_t) \\ \mathbf{h}^{A/O,*} &= \sum_{t=P_i^{(A/O,s)}}^{P_i^{(A/O,e)}} \alpha_t \cdot x_t \end{aligned} \quad (13)$$

where \mathbf{W}_3 and \mathbf{V} are parameters.

3) *High-Order Aggregation Layer*: After the prediction of aspect–opinion term pairs by the nonautoregressive decoder, we next perform the sentiment polarity classification for each pair. As we rendered in Section I, there is a high chance that the aspect and opinion terms overlap with each other. These overlapping aspect–opinion pairs can share rich interaction information, which may provide clues to predicting sentiment polarity. For instance the third sentence (c) in Fig. 1, all the aspects triggered by the opinion word “nice” hold the same positive sentiment, while those aspects under opinion expression “too steep price” share the same negative sentiment.

We here propose a high-order aggregation module for fully leveraging the underlying common information within the structures. Technically, we view each aspect–opinion pair [denoted as $\pi_i(A_*, O_*)$] as a node, and whenever a term (either the aspect or the opinion term) in π_i also cooccurs in other pair π_j , we assign an edge between these two nodes. In this way, we form an undirected graph $G = (V, E)$, where V is a set of pair nodes and E is a set of bidirectional edges between nodes. We first concatenate the representations of aspect terms and opinion terms as the pair node representation $\mathbf{r}_i^P = [\mathbf{r}_i^A; \mathbf{r}_i^O]$ and then employ a GCN encoder to model the graph G

$$\mathbf{g}_i = \sigma(\mathbf{W}_4 \mathbf{r}_i^P + b) \quad (14)$$

$$\mathbf{r}_i^{P,*} = \text{ReLU}\left(\sum_{j \in \mathcal{N}(i)} \mathbf{r}_j^P \odot \mathbf{g}_i\right) \quad (15)$$

where $\mathcal{N}(i)$ are neighbors of the node π_i and \mathbf{W} is the parameter. \odot is an elementwise multiplication. σ and ReLU are nonlinear activation functions. Thereafter, we put a softmax layer on the resulting pair representation, i.e., $c = \text{Softmax}(\mathbf{r}_i^{P,*})$, to generate the final sentiment label for the i th triplet.

C. Training

Our training target is to narrow the gap between the gold annotations with the predicted ones. As mentioned earlier, the cross-entropy loss is sensitive to the permutation of the predictions, which does not fit the nonautoregressive decoding of unordered set prediction. We consider a loss function for generating the optimal bipartite matching between predicted and ground-truth triples. Let us review the predicted triplet structure $Y = \{y_i\}_{i=1}^D$ with indexes

$$y_i = \left\{ \left(\mathbf{P}_i^{(A,s)}, \mathbf{P}_i^{(A,e)} \right), \left(\mathbf{P}_i^{(O,s)}, \mathbf{P}_i^{(O,e)} \right), \mathbf{c}_i \right\} \quad (16)$$

where \mathbf{P} is the distribution of its position index P and \mathbf{c} is the distribution of the label id c . We then write the gold one as

$$\hat{y}_i = \left\{ \left(\hat{\mathbf{P}}_i^{(A,s)}, \hat{\mathbf{P}}_i^{(A,e)} \right), \left(\hat{\mathbf{P}}_i^{(O,s)}, \hat{\mathbf{P}}_i^{(O,e)} \right), \hat{\mathbf{c}}_i \right\} \quad (17)$$

in the gold set $\hat{Y} = \{\hat{y}_i\}_{i=1}^K$, where $K \ll D$. We align the length of gold triplet sets \hat{Y} with the predicted one Y (i.e., into the size of D) by padding \hat{Y} with empty triplets that comprise dummy positions and null sentiment labels, $\phi = \{(\varepsilon, \varepsilon), (\varepsilon, \varepsilon), \varepsilon\}$. Next, we divide the loss calculation into two steps: 1) searching for an optimal matching between the gold triple set and the predicted triple set and 2) computing the loss between the matched pairs.

TABLE I

STATISTICS OF DATASETS. NOTE THAT # DENOTES THE NUMBER OF THE ITEMS. *Sent.*, *Asp.*, *Opn.*, AND *Ovlp* ARE THE ABBREVIATIONS OF “SENTENCE,” “ASPECT,” “OPINION,” AND “OVERLAPPING,” RESPECTIVELY. *Avg.Asp.Len* AND *Avg.Opn.Len* INDICATE THE AVERAGE LENGTH OF THE ASPECT AND OPINION TERMS, RESPECTIVELY, WHILE *Avg.Pair.Dist* IS THE AVERAGE DISTANCE BETWEEN THE TERMS IN ONE PAIR

| Dataset | #Sent. | | | #Asp. | | #Opn. | | #Asp.-Opn. Pair | | |
|---------|--------|-----|------|-------|-------------|-------|-------------|-----------------|---------------|---------------|
| | Train | Dev | Test | #All | Avg.Asp.Len | #All | Avg.Opn.Len | #All | #Ovlp | Avg.Pair.Dist |
| 14res | 1,300 | 323 | 496 | 3,399 | 1.40 | 3,443 | 1.16 | 3,909 | 1,615 (41.3%) | 2.40 |
| 14lap | 920 | 228 | 339 | 2,040 | 1.47 | 2,046 | 1.25 | 2,349 | 1,009 (43.0%) | 2.75 |
| 15res | 593 | 148 | 318 | 1,507 | 1.46 | 1,638 | 1.19 | 1,747 | 617 (35.3%) | 2.26 |
| 16res | 842 | 210 | 320 | 1,946 | 1.44 | 2,101 | 1.19 | 2,247 | 788 (35.1%) | 2.22 |

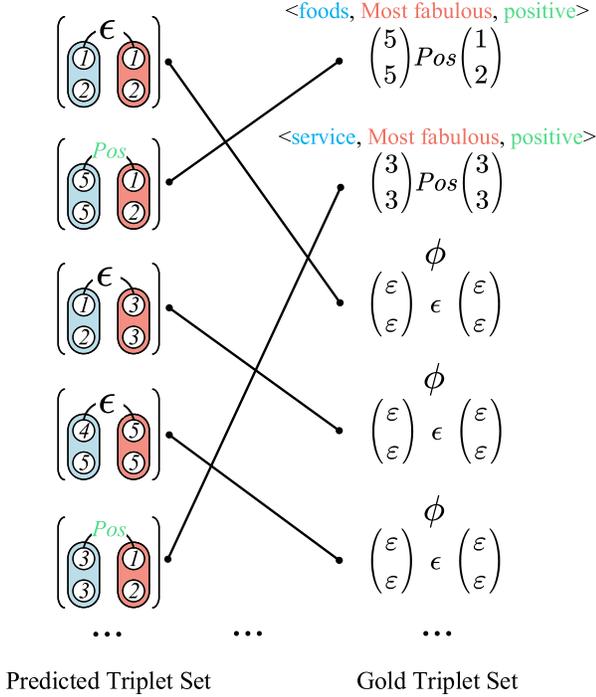


Fig. 4. Illustration of the bipartite matching between gold triplet set and predicted triplet set.

1) *Matching*: We aim to find a permutation of elements ψ^* with the lowest cost via the following matching mechanism:

$$\psi^* = \underset{\psi_i \in \Psi(D)}{\text{Argmin}} \sum_{i=1}^D \text{Scoring}(y_i, \hat{y}_{\psi_i}) \quad (18)$$

where $\Psi(D)$ is the complete space of D -size permutation. The distance between the gold \hat{y}_i and the predicted one y_{ψ_i} is measured via scoring function

$$\begin{aligned} \text{Scoring}(y_i, \hat{y}_{\psi_i}) = & -\mathbb{1}_{c_i \neq \epsilon} \left(\hat{c}_i \otimes \mathbf{c}_i + \hat{P}_i^{A,s} \otimes \mathbf{P}_i^{A,s} \right. \\ & + \hat{P}_i^{A,e} \otimes \mathbf{P}_i^{A,e} + \hat{P}_i^{O,s} \otimes \mathbf{P}_i^{O,s} \\ & \left. + \hat{P}_i^{O,e} \otimes \mathbf{P}_i^{O,e} \right) \end{aligned} \quad (19)$$

where \otimes refers to the operation of the summation after elementwise multiplication between the distributional item (i.e., \mathbf{P}) and the one-hot representation of the scalar item (i.e., \hat{P}). The searching can be satisfied by the Hungarian algorithm [76]. Fig. 4 shows the bipartite matching step.

2) *Computing Loss*: With the best matching pairs between the gold and predicted triplets, we next compute the loss

$$\begin{aligned} \mathcal{L}(Y, \hat{Y}) = & - \sum_{i=1}^D \left(\log \hat{c}_i \otimes \mathbf{c}_{\psi_i^*} + \log \hat{P}_i^{A,s} \otimes \mathbf{P}_{\psi_i^*}^{A,s} \right. \\ & + \log \hat{P}_i^{A,e} \otimes \mathbf{P}_{\psi_i^*}^{A,e} + \log \hat{P}_i^{O,s} \otimes \mathbf{P}_{\psi_i^*}^{O,s} \\ & \left. + \log \hat{P}_i^{O,e} \otimes \mathbf{P}_{\psi_i^*}^{O,e} \right) \end{aligned} \quad (20)$$

where ψ_i^* is the optimal matching index of the i th triplet via (18).

V. EXPERIMENTAL SETUP

A. Datasets

The experiments are performed based on the ASTE benchmark datasets, assembled by Peng *et al.* [8]. The data are derived from SemEval tasks [28], including *2014 restaurant* (14res), *2014 laptop* (14lap), *2015 restaurant* (15res), and *2016 restaurant* (16res). Each dataset comes with its own training set, development set, and test set. To make a fair comparison with existing baselines, we follow the same data preprocessing in [8]. The detailed statistics of the four datasets are listed in Table I.

B. Baseline Methods

To verify the effectiveness of our proposed framework, we make comparisons with the previous state-of-the-art methods designed for sentiment-oriented extraction tasks. We divide the baselines into two groups, namely, pipeline methods and end-to-end methods.

1) *Pipeline Models*: The models are given in the following.

1) *CLMA+* is a revision of CLMA model [1], which is constructed for aspect and opinion terms coextraction by leveraging the attention mechanism.

Peng *et al.* [8] revised the model as the first stage of terms extraction and then determined the sentiment polarity at the second stage to form triplets.

2) *RINANTE+*: Peng *et al.* [8], following the same way as of CLMA+ model, upgraded the RINANTE model [9] that performs term coextraction for ASTE.

3) *Li-unified-R+* is derived from the Li-unified-R model [12], which is a unified model that contains two-layer stacked long short-term memory (LSTM) for ATE and sentiment polarity classification. Peng *et al.* [8] modified Li-unified-R as Li-unified-R+ to extract opinion terms and then make pairing to obtain sentiment triplets.

TABLE II
RESULTS ON FOUR DATASETS. SCORES IN GRAY COLOR BACKGROUND ARE FROM OUR REIMPLEMENTATION, WHILE OTHERS ARE RETRIEVED FROM THE CORRESPONDING RAW PAPERS WITHOUT MODIFICATION

| | | AE | | | AE+SC | | | OE | | | AOP | | | ASTE | | |
|---------------------|---------------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|
| | | P | R | F1 |
| • 14res data | | | | | | | | | | | | | | | | |
| Pipeline | RINANTE+ | 75.89 | 70.34 | 73.00 | 48.97 | 47.36 | 48.15 | 81.06 | 72.05 | 76.29 | 42.32 | 51.08 | 46.29 | 31.07 | 37.63 | 34.03 |
| | CMLA+ | 84.21 | 89.83 | 86.93 | 67.80 | 73.69 | 70.62 | 69.47 | 74.53 | 71.91 | 45.17 | 53.42 | 48.95 | 40.11 | 46.63 | 43.12 |
| | Li-unified-R+ | - | - | - | 73.15 | 74.44 | 73.79 | 81.20 | 83.18 | 82.13 | 44.37 | 73.67 | 55.34 | 41.44 | 68.79 | 51.68 |
| | TSM | - | - | - | 74.41 | 73.97 | 74.19 | 81.77 | 84.80 | 83.21 | 47.76 | 68.10 | 56.10 | 44.18 | 62.99 | 51.89 |
| | HSLM | - | - | 79.50 | - | - | - | - | - | 81.00 | - | - | 73.00 | 71.73 | 64.15 | 67.73 |
| End-to-end | JET | 83.02 | 85.30 | 82.36 | 71.36 | 73.31 | 72.68 | 80.66 | 79.89 | 80.35 | 72.30 | 72.75 | 72.60 | 66.76 | 49.09 | 56.58 |
| | GTS | 80.34 | 83.69 | 81.20 | 74.12 | 75.50 | 74.63 | 82.08 | 85.36 | 83.34 | 74.13 | 69.49 | 71.74 | 70.79 | 61.71 | 65.94 |
| | BMRC | 85.64 | 86.56 | 85.82 | 74.73 | 74.31 | 73.52 | 83.12 | 81.56 | 82.55 | 74.50 | 72.48 | 73.55 | 71.31 | 63.52 | 66.47 |
| | Ours | 85.82 | 90.54 | 88.15 | 76.33 | 76.85 | 76.45 | 82.36 | 85.60 | 85.11 | 75.63 | 73.32 | 74.91 | 72.84 | 65.30 | 68.82 |
| | • 14lap data | | | | | | | | | | | | | | | |
| Pipeline | RINANTE+ | 70.80 | 52.80 | 60.50 | 41.20 | 33.20 | 36.70 | 78.20 | 62.70 | 69.60 | 34.40 | 26.20 | 29.70 | 23.10 | 17.60 | 20.00 |
| | CMLA+ | 71.50 | 82.20 | 76.40 | 54.70 | 59.20 | 56.90 | 51.80 | 65.30 | 57.70 | 42.10 | 46.30 | 44.10 | 31.40 | 34.60 | 32.90 |
| | Li-unified-R+ | - | - | - | 66.28 | 60.71 | 63.38 | 76.62 | 74.90 | 75.70 | 52.29 | 52.94 | 52.56 | 42.25 | 42.78 | 42.47 |
| | TSM | - | - | - | 64.35 | 60.29 | 62.26 | 76.87 | 75.31 | 76.03 | 50.00 | 58.47 | 53.85 | 40.40 | 47.24 | 43.50 |
| | HSLM | - | - | 75.30 | - | - | - | - | - | 70.80 | - | - | 58.00 | 52.35 | 47.76 | 49.95 |
| End-to-end | JET | 70.35 | 78.65 | 73.35 | 63.03 | 62.30 | 62.85 | 75.36 | 76.70 | 75.82 | 65.02 | 56.45 | 58.68 | 52.00 | 35.91 | 42.48 |
| | GTS | 71.66 | 80.13 | 76.31 | 65.35 | 61.08 | 63.72 | 77.85 | 77.02 | 77.36 | 68.33 | 55.04 | 60.97 | 55.93 | 47.52 | 51.38 |
| | BMRC | 71.11 | 81.52 | 76.12 | 63.41 | 61.84 | 62.27 | 76.33 | 77.76 | 77.12 | 67.12 | 56.32 | 61.22 | 54.12 | 44.44 | 50.50 |
| | Ours | 72.33 | 82.34 | 77.68 | 65.86 | 63.69 | 64.77 | 78.60 | 78.75 | 78.55 | 69.61 | 58.52 | 62.35 | 56.38 | 48.78 | 53.76 |
| | • 15res data | | | | | | | | | | | | | | | |
| Pipeline | RINANTE+ | 72.64 | 51.68 | 60.39 | 46.20 | 37.40 | 41.30 | 77.40 | 57.00 | 65.70 | 37.10 | 33.90 | 35.40 | 29.40 | 26.90 | 28.00 |
| | CMLA+ | 75.10 | 89.30 | 81.50 | 49.90 | 58.00 | 53.60 | 60.80 | 65.30 | 62.90 | 42.70 | 46.70 | 44.60 | 34.40 | 37.60 | 35.90 |
| | Li-unified-R+ | - | - | - | 64.95 | 64.95 | 64.95 | 79.18 | 75.88 | 77.44 | 52.75 | 61.75 | 56.85 | 43.34 | 50.73 | 46.69 |
| | TSM | - | - | - | 63.41 | 65.19 | 64.29 | 75.98 | 76.32 | 76.10 | 49.22 | 65.70 | 56.23 | 40.97 | 54.68 | 46.79 |
| | HSLM | - | - | 75.50 | - | - | - | - | - | 73.00 | - | - | 65.00 | 62.37 | 54.29 | 58.05 |
| End-to-end | JET | 73.68 | 82.68 | 75.36 | 64.51 | 62.38 | 63.30 | 73.33 | 70.69 | 72.65 | 64.88 | 62.30 | 63.35 | 59.77 | 42.27 | 49.52 |
| | GTS | 76.24 | 85.30 | 77.42 | 66.18 | 64.72 | 65.65 | 76.85 | 74.98 | 75.10 | 66.26 | 63.19 | 65.39 | 60.09 | 53.57 | 56.64 |
| | BMRC | 75.55 | 84.52 | 77.08 | 65.31 | 61.52 | 63.15 | 78.12 | 75.58 | 77.00 | 67.12 | 60.19 | 63.51 | 62.41 | 54.76 | 57.83 |
| | Ours | 77.19 | 88.30 | 82.64 | 67.55 | 65.85 | 66.41 | 80.35 | 78.36 | 78.69 | 66.85 | 66.05 | 66.61 | 63.58 | 55.83 | 60.25 |
| | • 16res data | | | | | | | | | | | | | | | |
| Pipeline | RINANTE+ | 67.10 | 55.20 | 60.60 | 49.40 | 36.70 | 42.10 | 75.00 | 42.40 | 54.10 | 35.70 | 27.00 | 30.70 | 27.10 | 20.50 | 23.30 |
| | CMLA+ | 72.00 | 87.60 | 79.00 | 58.90 | 63.60 | 61.20 | 74.50 | 69.00 | 71.70 | 52.50 | 47.90 | 50.00 | 43.60 | 39.80 | 41.60 |
| | Li-unified-R+ | - | - | - | 66.33 | 74.55 | 70.20 | 79.84 | 86.88 | 83.16 | 46.11 | 64.55 | 53.75 | 38.19 | 53.47 | 44.51 |
| | TSM | - | - | - | 69.74 | 71.62 | 70.67 | 82.33 | 85.16 | 83.67 | 52.35 | 70.50 | 60.04 | 46.76 | 62.97 | 53.62 |
| | HSLM | - | - | 79.50 | - | - | - | - | - | 84.00 | - | - | 72.30 | 67.54 | 66.24 | 66.88 |
| End-to-end | JET | 70.35 | 80.61 | 76.18 | 68.51 | 70.13 | 69.20 | 80.25 | 84.32 | 82.68 | 68.64 | 70.48 | 69.31 | 63.59 | 50.97 | 56.59 |
| | GTS | 75.16 | 86.38 | 79.85 | 70.68 | 71.45 | 71.05 | 83.25 | 85.64 | 84.32 | 70.48 | 72.39 | 71.42 | 62.63 | 66.98 | 64.73 |
| | BMRC | 74.12 | 87.43 | 79.70 | 70.84 | 70.11 | 70.46 | 82.50 | 84.62 | 83.71 | 69.12 | 72.14 | 70.91 | 66.16 | 64.71 | 65.52 |
| | Ours | 76.84 | 89.45 | 81.49 | 72.81 | 72.05 | 72.29 | 84.60 | 86.02 | 85.65 | 70.89 | 74.51 | 73.12 | 68.35 | 67.28 | 68.03 |

4) *Two-Stage Model (TSM)*: Peng *et al.* [8] proposed a TSM for sentiment triplet extraction, i.e., from aspect-opinion term coextraction to term pairing.

2) *End-to-End Models*: The models are given in the following.

- 1) *HSLM*: Chen *et al.* [16] introduced a hierarchical sequence labeling model to tag the aspect terms, opinion expressions, and sentiment polarity in a joint manner.
- 2) *JET*: Xu *et al.* [17] proposed a position-aware joint model based on sequential labeling for ASTE.
- 3) GTS is a table-filling framework proposed by Wu *et al.* [18], which searches out the aspect terms and opinion expressions together with the correlated polarities in the rows and columns of the table.
- 4) *BMRC*: Chen *et al.* [56] transformed the triplet prediction into a machine reading comprehension task and solved it with an end-to-end manner. The method brings new state-of-the-art results to this task.

C. Implementations

We use 300-D word embeddings initialized with pretrained Glove [77]. The triplet query and position embedding are

randomly initialized with 300-D and 30-D, respectively. The convolutions in character CNN are with window sizes [3, 4, 5], each consisting of ten filters. The transformer encoder and decoder are all with three layers, having the default hidden size of 768-D. All hidden sizes of the rest representations are set with 250-D. The GCN in the high-order aggregation layer is set with two layers. The maximum triplet length D is set as 8,² which is a tradeoff between effectiveness and efficiency by our preliminary experiments. We use mini-batch with a size of 16, training with unlimited iterations but with early stopping strategy. We adopt the Adam optimizer with an initial learning rate as $1e^{-4}$ and an L2 weight decay of $5e^{-5}$. The environment is with Intel i9 CPU and NVIDIA GeForce RTX 3090 GPU card with 11-GB graphic memory. The implementation of our system is with the PyTorch library.³

D. Evaluations

We use precise (P), recall (R), and F1 score (F1) to measure the performance. Note that a predicted triplet is correct

²According to our statistics, the maximum triplet size in all sentences is less than 8.

³<https://pytorch.org/>

only if aspect term, opinion term, and sentiment polarity are all correct at the same time. We tune our framework on the development sets for obtaining the best hyperparameters, which are then adopted on the corresponding test sets. Each of our model is trained ten times and all the scores are presented after the significance test ($p \leq 0.03$). We also verify the performances by using the pretrained contextualized BERT language model⁴ [78], [79], and the ELMo language model⁵ [80].

VI. RESULTS AND DISCUSSION

A. Main Results

We perform evaluations on the ASTE task and the satellite subtasks concerning ASTE, i.e., aspect extraction (AE), SC, opinion extraction (OE), and aspect and opinion pair extraction (AOP). Experimental results are presented in Table II, from which we can correspondingly gain several observations.

First, the performances of ASTE are much better by the end-to-end (or joint) models than those by the pipeline methods, across all the four datasets. This corresponds much to the prior research findings [16]–[18], as the joint prediction of the subtasks can greatly ease the error propagation issues existed in the cascade procedure. We notice that such performance improvements are not significant for standalone jobs, such as AE and OE. This verifies the longstanding viewpoint that the joint tasks can be best accomplished with the joint modeling, rather than the pipeline methods [8], [11], [12].

Second, we look into the variances of the performances among different combinations of the tasks, finding that the performances of the satellite subtasks within the ASTE task by the end-to-end models are significantly higher than that of the separate subtasks by the pipeline methods. For example, the F1 scores of the AE + SC task by the best pipeline method (i.e., TSM) are 74.19%, 62.26%, 64.29%, and 70.67%, on the four datasets. By simultaneously modeling the OE subtask, i.e., with the model capturing the additional opinion information, the F1 scores of the same AE + SC task by the best joint method (i.e., GTS) are increased to 74.63%, 63.72%, 65.65%, and 71.05% on each dataset. This can be viewed as the direct evidence to support the usefulness of the ASTE task that integrates the satellite subtasks into a complete job, introducing additional sentiment-relevant clues for facilitating the overall sentiment extraction.

Most importantly, our proposed framework outperforms all the competitors with considerable gaps. We see that the winning scores by our model against the baselines are consistently preserved in all the satellite subtasks among all the datasets. In brief, our model achieves the overall 68.82%, 53.76%, 60.25%, and 68.03% F1 scores on each dataset. This proves the effectiveness of our proposed pointer-net-based nonautoregressive framework for ASTE task. Besides, the contributions for the F1 score improvements are derived from both the higher precision rate and recall rate, simultaneously, which demonstrates the advantages of our framework. On the one hand, our neural pointer-net-based architecture

⁴<https://github.com/google-research/bert>, uncased base version.

⁵<https://allennlp.org/elmo>

TABLE III

ABLATION RESULTS ON OUR FRAMEWORK. VALUES ARE IN F1 SCORE FOR ASTE TASK. “LSTM DECODER + CE LOSS” MEANS USING A SEQUENTIAL LSTM AS THE DECODER AND REPLACING THE BIPARTITE MATCHING LOSS WITH A CROSS-ENTROPY LOSS (I.E., IN ORDERED TRIPLET PREDICTION). “HO AGGREGATION” REFERS TO THE HIGH-ORDER AGGREGATION MECHANISM

| | 14res | 14lap | 15res | 16res | Avg. |
|--------------------|--------------|--------------|--------------|--------------|--------------|
| Ours (full model) | 68.82 | 53.76 | 60.25 | 68.03 | 62.72 |
| Xavier Emb. | 68.30 | 53.56 | 59.53 | 67.80 | 62.29 |
| w/o Enc. Pos. Emb. | 68.05 | 52.98 | 59.68 | 67.85 | 62.14 |
| w/o Char Emb. | 68.36 | 53.61 | 59.75 | 67.91 | 62.40 |
| w/o Dec. Pos. Emb. | 68.27 | 53.31 | 59.84 | 67.97 | 62.35 |
| BiLSTM encoder | 68.58 | 53.02 | 59.35 | 67.66 | 62.15 |
| LSTM decoder | 67.92 | 52.65 | 58.81 | 67.40 | 61.69 |
| + CE Loss | 67.56 | 51.89 | 58.08 | 67.12 | 61.16 |
| w/o HO aggregation | 67.78 | 52.22 | 58.75 | 67.30 | 61.51 |
| Asp.-edge graph | 67.78 | 52.22 | 58.75 | 67.30 | 61.51 |
| Opn.-edge graph | 67.78 | 52.22 | 58.75 | 67.30 | 61.51 |
| +ELMo | 70.49 | 54.28 | 60.74 | 69.23 | 63.68 |
| +BERT | 71.37 | 55.07 | 61.83 | 70.06 | 64.58 |

contributes to the high precision, while on the other hand, the nonautoregressive framework with unordered set prediction by the bipartite matching loss results in the high recall rate.

B. Model Ablation

We perform the ablation study to assess the contribution of each aspect in our method. We mainly focus on four aspects: 1) input features; 2) model architecture design; 3) the proposed high-order aggregation module; and 4) the effectiveness of the external contextualized word representations. Table III shows the experimental results.

We first replace the pretrained word embedding GloVe with the randomly initialized one via the Xavier algorithm [81]. The overall performance drops. Also, without neither the character representations from CNN nor the position embeddings, the results are worse, interestingly with the position information (both on encoding and decoding side) showing more impacts to our system. This can be intuitive because the position clues contribute more to the pointer-net-based system.

Next, we replace the transformer encoder with a bidirectional LSTM for computational linguistics (BiLSTM) encoder (in our practice, we fine-tune the performances with the best configurations) and find slight performance drops. We then use the sequential-like LSTM as our decoder for yielding the triplets (where it turns to be the autoregressive decoding), and we see that the system starts performing poorly, with an average drop of 1.03% (62.72–61.69) F1 score. Based on the LSTM decoder architecture, we further substitute the bipartite matching loss with cross-entropy loss. The overall performance drops grow to averaged 1.56% (62.72–61.16) F1 score. This proves the effectiveness of our nonautoregressive framework with the bipartite matching loss for unordered set prediction.

We ablate the high-order aggregation mechanism and find that the overall performances decrease significantly, e.g., averaged 1.21% (62.72–61.51) F1 drop, which demonstrates the effectiveness of the proposed mechanism. In the high-order

aggregation module, we construct edges in the graph by checking whether either the aspect term or the opinion term cooccurs in other pair. Here, we take a further step, exploring the influences if we form the edges depending only on the aspect term overlapping or only on the opinion term overlapping. We find that the performances in either standalone “Asp.-edge graph” or “Opn.-edge graph” are inferior to that in the complete graph. Also, we see that the graph with edges from only the overlapping opinion terms is more beneficial to the task by the counterpart from the overlapping aspect terms.

Finally, we equip our framework with the contextualized language models, e.g., ELMo and BERT. Unsurprisingly, we receive significant performance gains, i.e., with averaged 63.68% by ELMo and 64.58% BERT. The tendencies are quite coincident with the recent findings that employ the contextualized language models, which can lead to large task improvements [18], [78].

C. Analysis

1) *Overlapping Triplet Extraction*: As we emphasized earlier, the triplet overlaps much with each other on the datasets, which largely influences the overall results of ASTE. Now, we study to how our framework can better fight against the triplet overlapping phenomenon. We conduct experiments based on the four datasets and make comparisons with baselines. We show the results in Fig. 5. Our framework achieves significant improvements over the baseline systems on handling the overlapping issues. Specifically, almost all the baselines fail to recognize the overlapping triplets when the overlapping number is 6, while our model gives comparatively well results. When the bipartite matching loss is unavailable and the design of unordered triplet set prediction shifts to the standard pointer network-based extraction model, we correspondingly find the comparative and clear performance drops. This reveals the usefulness of our proposed framework. Note that even with the standard pointer network-based framework, we still outperform the baselines by a large margin.

2) *Influence of Term Span Length*: In an extraction task, longer term spans involve more varying boundary tokens, which correspondingly brings more challenges to the extractor. We here study the impacts of different lengths of term spans by different models. We perform experiments on the 14res test set, and the results are shown in Fig. 6. In general, the performances of all the models decrease when the lengths of span terms grow. The joint methods show stronger capabilities than the pipeline model (i.e., TSM) on handling the increase of span widths. However, we see that the table-filling-based joint model GTS performs consistently better than the sequential-labeling model JET. More importantly, our proposed framework outperforms all the baselines. Meanwhile, the improvements become more significant even when the span length goes larger. This reflects the ability of our model that employs the pointer network as the extraction backbone.

3) *Impacts of the Distance Between Aspect and Opinion Terms*: Long-range dependencies are a longstanding problem in information extraction. When two observed mentions (here the aspect and opinion terms) are separated farther away,

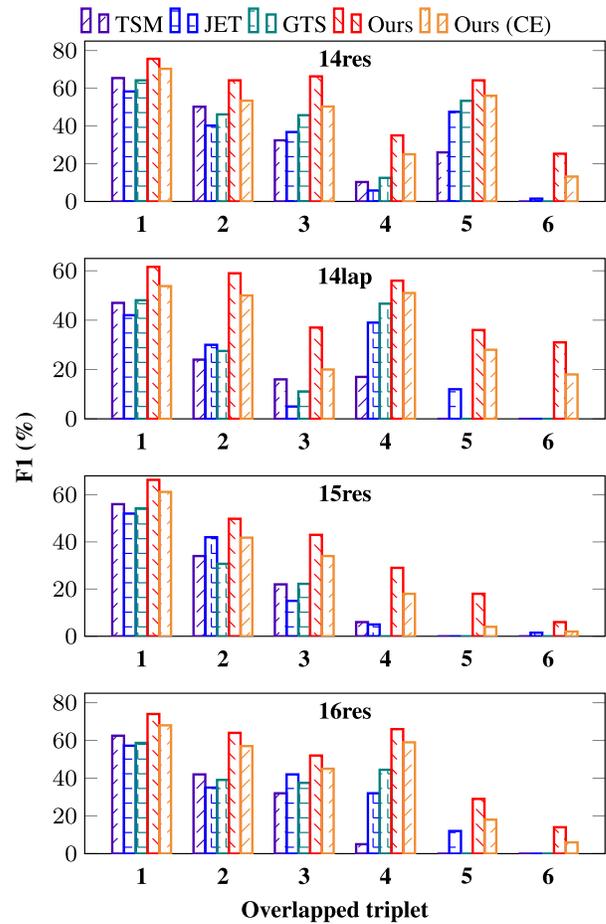


Fig. 5. Comparisons on different overlapping triplets. “Our(CE)” refers to our framework where the bipartite matching loss is replaced with cross-entropy loss, and the nonautoregressive transformer decoder is substituted with a sequential LSTM decoder.

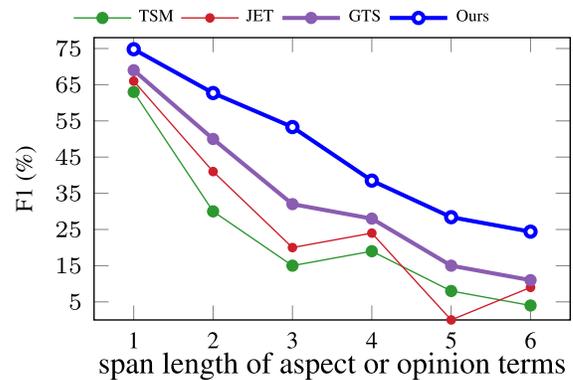


Fig. 6. Influence of aspect/opinion term span length.

the extraction becomes harder. We explore the effects of such long-range dependency issues. Technically, we vary the distances between aspect and opinion terms of the models and observe the change of performances. Fig. 7 shows the results based on the 14res test set. We see that the overall performances of all the models get worse once the distance increases. Our framework with the pointer mechanism can

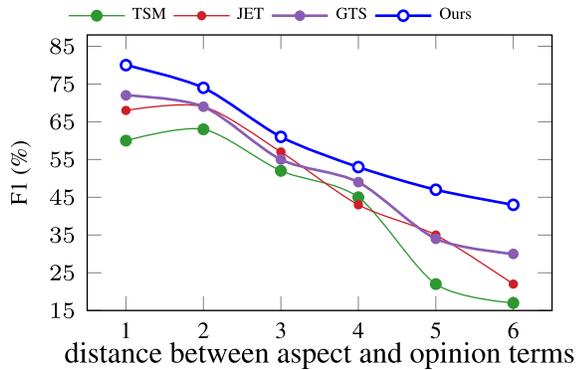


Fig. 7. Results in different aspect–opinion distance.

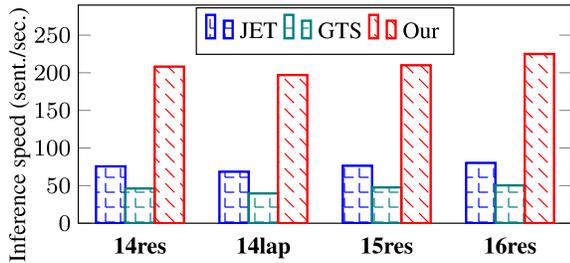


Fig. 8. Comparisons in terms of the decoding speed.

alleviate such performance descending. There are two major reasons. The first is the pointer network design, where each pointing decision is made by the consultant of all the input words at the global level. The second is the leverage of the transformer model as the decoder for the nonautoregressive prediction of unordered triplet set. The self-attention with multiple head mechanism in the transformer encoder enables it more effectively to retrieve informative features from bidirectional contexts.

4) *Decoding Efficiency*: One of the major advantages of our nonautoregressive encoder–decoder framework is its decoding efficiency. Theoretically, our framework yields all the possible results without relying on the input sequence length, giving $O(1)$ time consumption on decoding. Here, we study the model running speed during the decoding phase against the baseline joint models for ASTE. We present the inference speeds (i.e., sentence per second) of all the models in Fig. 8. Unsurprisingly, our framework shows the lowest time consumption on decoding compared with JET and GTS. We note that both JET and table-filling-based method GTS is much time-consuming, as it takes $O(n^2)$ time complexity to iteratively enumerate the aspect terms and the opinion terms for pairing them, while ours is time-independent.

5) *Training With Bipartite Matching Loss*: The main motivation of our proposal of the bipartite matching loss is to achieve the unordered triplet set prediction, along with the nonautoregressive decoding architecture. Here, we study the influence of the loss functions toward the training process. We mainly make a comparison between the bipartite matching loss and the cross-entropy loss. Fig. 9 plots the training curves on each of the four datasets. We correspondingly have two

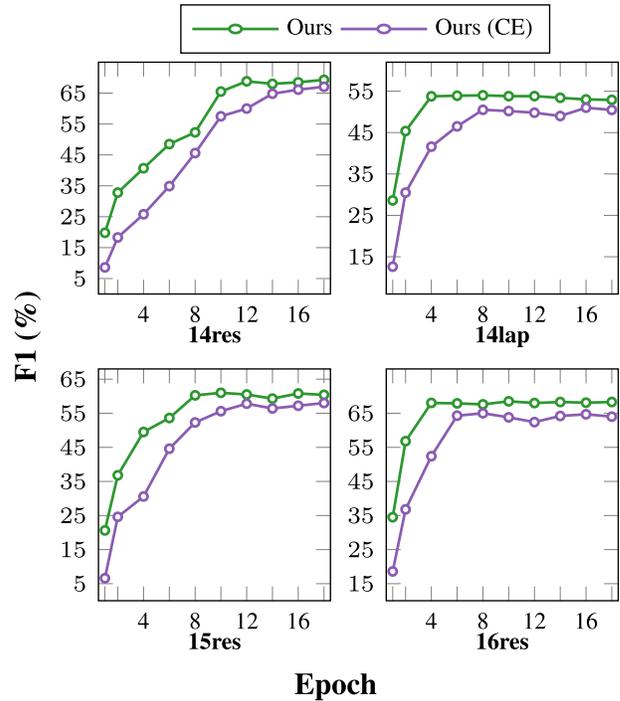


Fig. 9. Training curves on each dataset.

major observations. First, we see that the overall converging F1 results by the bipartite matching loss are higher than the results by the cross-entropy loss. Second, the times for the model to converge are overall shorter with bipartite matching loss than that with cross-entropy loss, which can be learned from the patterns in the figures.

D. Case Study

Finally, we perform a case study by comparing our model with two strong baselines JET and GTS. Fig. 10 shows the results of three different sentences, including the overlapping situation, based on the 16res data. First, we can find that, generally, our model yields the fully correct predictions to the gold-standard triplets. Note that two strong baseline systems either miss the prediction of several triplets or produce wrong opinion terms or incorrect sentiment polarities. For example, in the first and second sentences, both JET and GTS fail to recognize the full numbers of triplets. This is large because the architecture of the autoregressive modeling in baselines. In the second sentence, we find that the sequence-labeling-based model JET wrongly extracts the single token “busy” as opinion term and meanwhile recognizes incorrect pair between aspect–opinion terms (i.e., “margaritas” and “busy”). The GTS model can detect more triplets than JET, partially due to the table-filling modeling on exhaustively enumerating all possible aspect–opinion term pairs. However, it still fails to correctly predict all possible sentiment triplets. For the third sentence, two baselines commit these errors, where, however, our encoder–decoder-based model equipped with a pointer network can recognize all the triplets and precisely capture the negated opinion. Besides, both JET and GTS make mistakes frequently in determining sentiment polarity, while

| Test sentence | Gold triplet | JET | GTS | Ours |
|--|---|---|--|---|
| <i>I've had the Jellyfish, Horse Mackerel, Blue Fin Tuna and the Sake Ikura roll among others, and they were all good.</i> | (Jellyfish, good, positive) (Horse Mackerel, good, positive) (Blue Fin Tuna, good, positive) (Sake Ikura roll, good, positive) | (Jellyfish, good, positive) ✓ (Horse Mackerel, good, positive) ✓ | (Jellyfish, good, positive) ✓ (Horse Mackerel, good, positive) ✓ (Sake Ikura roll, good, positive) ✓ | (Jellyfish, good, positive) ✓ (Horse Mackerel, good, positive) ✓ (Blue Fin Tuna, good, positive) ✓ (Sake Ikura roll, good, positive) ✓ |
| <i>The food was great, the margaritas too, but the waitress was too busy.</i> | (food, great, positive) (margaritas, great, positive) (waitress, too busy, negative) | (food, great, positive) ✓ (margaritas, busy, negative) ✗ (waitress, busy, negative) ✗ | (food, great, positive) ✓ (waitress, too busy, negative) ✓ | (food, great, positive) ✓ (margaritas, great, positive) ✓ (waitress, too busy, negative) ✓ |
| <i>The food was excellent as well as service, however, I still left Michelin disappointed.</i> | (food, excellent, positive) (service, excellent, positive) (Michelin, disappointed, negative) | (food, excellent, positive) ✓ (service, disappointed, negative) ✗ | (food, excellent, positive) ✓ (service, excellent, positive) ✓ (Michelin, disappointed, positive) ✗ | (food, excellent, positive) ✓ (service, excellent, positive) ✓ (Michelin, disappointed, negative) ✓ |

Fig. 10. Results of case study by different models based on three test sentences randomly selected from the 16res data. Correct predictions are followed with green check marks and incorrect ones are with red cross marks.

our model shows high effectiveness on this. We give the major credit to our proposed high-order aggregation module, which can effectively exploit the shared information between the overlapped pairs of aspect and opinion terms.

VII. CONCLUSION AND FUTURE WORK

We present a novel framework for end-to-end ASTE based on the encoder–decoder architecture. We model the task as an unordered triplet set prediction problem and perform nonautoregressive decoding with the transformer-based pointer network. A high-order aggregation module is presented to fully explore the underlying interactions between aspect and opinion terms for SC. We also introduce a bipartite matching loss for better training our nonautoregressive system. Experimental results on four benchmark datasets show that our proposed framework significantly outperforms previous state-of-the-art methods. Ablation studies uncover the contribution of each part of the proposed framework. A list of analyses demonstrates the advantages of our model in multiple aspects, including stronger robustness on term span length, relieving long-range dependency issue, better handling the overlapping problem, and highly decoding efficiency, compared with the previous methods.

As future work, considering the significant advantages on such triplet task modeling, we believe that our proposed nonautoregressive encoder–decoder system can be extended to many other applications that follow the triplet extraction scheme, e.g., relation extraction and semantic role labeling, for better task performances. Besides, we think that the integration of external structural information will further facilitate the extraction of the triplets, e.g., syntactic constituency tree for enhancing the recognition of terms and syntactic dependency tree for better exploring the relation of term pairs.

REFERENCES

- [1] W. Wang, S. J. Pan, D. Dahlmeier, and X. Xiao, “Coupled multi-layer attentions for co-extraction of aspect and opinion terms,” in *Proc. AAAI*, 2017, pp. 3316–3322.
- [2] S. Wang, S. Mazumder, B. Liu, M. Zhou, and Y. Chang, “Target-sensitive memory networks for aspect sentiment classification,” in *Proc. ACL*, 2018, pp. 957–967.
- [3] H. Fei, Y. Ren, and D. Ji, “Implicit objective network for emotion detection,” in *Proc. 8th CCF Int. Conf. Natural Lang. Process. Chin. Comput. (NLPCC)*, 2019, pp. 647–659.
- [4] H. Peng, Y. Ma, Y. Li, and E. Cambria, “Learning multi-grained aspect target sequence for Chinese sentiment analysis,” *Knowl.-Based Syst.*, vol. 148, pp. 167–176, May 2018.
- [5] C. Sun, L. Huang, and X. Qiu, “Utilizing BERT for aspect-based sentiment analysis via constructing auxiliary sentence,” in *Proc. NAACL*, 2019, pp. 380–385.
- [6] N. Liu and B. Shen, “ReMemNN: A novel memory neural network for powerful interaction in aspect-based sentiment analysis,” *Neurocomputing*, vol. 395, pp. 66–77, Jun. 2020.
- [7] H. Fei, Y. Zhang, Y. Ren, and D. Ji, “Latent emotion memory for multi-label emotion classification,” in *Proc. AAAI Conf. Artif. Intell.*, 2020, pp. 7692–7699.
- [8] H. Peng, L. Xu, L. Bing, F. Huang, W. Lu, and L. Si, “Knowing what, how and why: A near complete solution for aspect-based sentiment analysis,” in *Proc. AAAI*, 2020, pp. 8600–8607.
- [9] H. Dai and Y. Song, “Neural aspect and opinion term extraction with mined rules as weak supervision,” in *Proc. ACL*, 2019, pp. 5268–5277.
- [10] Z. Chen and T. Qian, “Relation-aware collaborative learning for unified aspect-based sentiment analysis,” in *Proc. ACL*, 2020, pp. 3685–3694.
- [11] Y. Tay, L. A. Tuan, and S. C. Hui, “Learning to attend via word-aspect associative fusion for aspect-based sentiment analysis,” in *Proc. AAAI*, 2018, pp. 5956–5963.
- [12] X. Li, L. Bing, P. Li, and W. Lam, “A unified model for opinion target extraction and target sentiment prediction,” in *Proc. AAAI*, 2019, pp. 6714–6721.
- [13] Y. Zhou, L. Huang, T. Guo, J. Han, and S. Hu, “A span-based joint model for opinion target extraction and target sentiment classification,” in *Proc. IJCAI*, 2019, pp. 5485–5491.
- [14] B. Wang and W. Lu, “Learning latent opinions for aspect-level sentiment classification,” in *Proc. AAAI*, 2018, pp. 5537–5544.
- [15] O. De Clercq, E. Lefever, G. Jacobs, T. Carpels, and V. Hoste, “Towards an integrated pipeline for aspect-based sentiment analysis in various domains,” in *Proc. 8th Workshop Comput. Approaches Subjectivity, Sentiment Social Media Anal.*, 2017, pp. 136–142.
- [16] P. Chen, S. Chen, and J. Liu, “Hierarchical sequence labeling model for aspect sentiment triplet extraction,” in *Proc. NLPCC*, 2020, pp. 654–666.
- [17] L. Xu, H. Li, W. Lu, and L. Bing, “Position-aware tagging for aspect sentiment triplet extraction,” in *Proc. EMNLP*, 2020, pp. 2339–2349.
- [18] Z. Wu, C. Ying, F. Zhao, Z. Fan, X. Dai, and R. Xia, “Grid tagging scheme for aspect-oriented fine-grained opinion extraction,” in *Proc. Findings Assoc. Comput. Linguistics, EMNLP*, 2020, pp. 2576–2585.
- [19] B. Wang, W. Lu, Y. Wang, and H. Jin, “A neural transition-based model for nested mention recognition,” in *Proc. EMNLP*, 2018, pp. 1011–1017.
- [20] H. Lin, Y. Lu, X. Han, and L. Sun, “Sequence-to-nuggets: Nested entity mention detection via anchor-region networks,” in *Proc. ACL*, 2019, pp. 5182–5192.
- [21] R. McDonald and J. Nivre, “Analyzing and integrating dependency parsers,” *Comput. Linguistics*, vol. 37, no. 1, pp. 197–230, Mar. 2011.
- [22] Z. Zhang, H. Zhao, and L. Qin, “Probabilistic graph-based dependency parsing with convolutional neural network,” in *Proc. ACL*, 2016, pp. 1382–1392.
- [23] S. Kurita and A. Søgaard, “Multi-task semantic dependency parsing with policy gradient for learning easy-first strategies,” in *Proc. ACL*, 2019, pp. 2420–2430.
- [24] I. Sutskever, O. Vinyals, and Q. V. Le, “Sequence to sequence learning with neural networks,” in *Proc. NIPS*, 2014, pp. 3104–3112.
- [25] A. Vaswani *et al.*, “Attention is all you need,” in *Proc. NIPS*, 2017, pp. 5998–6008.
- [26] H. Fei, Y. Ren, S. Wu, B. Li, and D. Ji, “Latent target-opinion as prior for document-level sentiment classification: A variational approach from fine-grained perspective,” in *Proc. Web Conf.*, Apr. 2021, pp. 553–564.

- [27] R. Varghese and M. Jayasree, "Aspect based sentiment analysis using support vector machine classifier," in *Proc. ICACCI*, 2013, pp. 1581–1586.
- [28] M. Pontiki, D. Galanis, J. Pavlopoulos, H. Papageorgiou, I. Androutsopoulos, and S. Manandhar, "SemEval-2014 task 4: Aspect based sentiment analysis," in *Proc. SemEval*, 2014, pp. 27–35.
- [29] J. Xu, B. Tang, X. Liu, and F. Xiong, "Automatic aspect-based sentiment analysis (AABSA) from customer reviews," in *Proc. AAAI*, 2020, pp. 47–66.
- [30] J. Shen *et al.*, "Dual memory network model for sentiment analysis of review text," *Knowl.-Based Syst.*, vol. 188, Jan. 2020, Art. no. 105004.
- [31] F. Ghorbanian and M. Jalali, "A novel large-scale model for real time sentiment analysis using online shop reviews and comments," in *Proc. 6th Int. Conf. Comput. Artif. Intell.*, Apr. 2020, pp. 515–518.
- [32] L. Dong, F. Wei, C. Tan, D. Tang, M. Zhou, and K. Xu, "Adaptive recursive neural network for target-dependent Twitter sentiment classification," in *Proc. ACL*, 2014, pp. 49–54.
- [33] Y. Ren, Y. Zhang, M. Zhang, and D. Ji, "Context-sensitive Twitter sentiment classification using neural network," in *Proc. AAAI*, 2016, pp. 215–221.
- [34] T. H. Nguyen and K. Shirai, "PhraseRNN: Phrase recursive neural network for aspect-based sentiment analysis," in *Proc. EMNLP*, 2015, pp. 2509–2514.
- [35] D. Tang, B. Qin, and T. Liu, "Aspect level sentiment classification with deep memory network," in *Proc. EMNLP*, 2016, pp. 214–224.
- [36] P. Lin, M. Yang, and J. Lai, "Deep mask memory network with semantic dependency and context moment for aspect level sentiment classification," in *Proc. IJCAI*, 2019, pp. 5088–5094.
- [37] W. Wang, S. J. Pan, D. Dahlmeier, and X. Xiao, "Recursive neural conditional random fields for aspect-based sentiment analysis," in *Proc. EMNLP*, 2016, pp. 616–626.
- [38] H. Fei, D. Ji, Y. Zhang, and Y. Ren, "Topic-enhanced capsule network for multi-label emotion classification," *IEEE/ACM Trans. Audio, Speech, Language Process.*, vol. 28, pp. 1839–1848, 2020.
- [39] T. Mullen and N. Collier, "Sentiment analysis using support vector machines with diverse information sources," in *Proc. EMNLP*, 2004, pp. 412–418.
- [40] T. Wilson, J. Wiebe, and P. Hoffmann, "Recognizing contextual polarity in phrase-level sentiment analysis," in *Proc. EMNLP*, 2005, pp. 347–354.
- [41] K. Sadamitsu, S. Sekine, and M. Yamamoto, "Sentiment analysis based on probabilistic models using inter-sentence information," in *Proc. LREC*, 2008, pp. 1–6.
- [42] T. Phienthrakul, B. Kijssirikul, H. Takamura, and M. Okumura, "Sentiment classification with support vector machines and multiple kernel functions," in *Proc. ICONIP*, 2009, pp. 583–592.
- [43] X. Li, L. Bing, W. Lam, and B. Shi, "Transformation networks for target-oriented sentiment classification," in *Proc. ACL*, 2018, pp. 946–956.
- [44] M. Hu, Y. Peng, Z. Huang, D. Li, and Y. Lv, "Open-domain targeted sentiment analysis via span-based extraction and classification," in *Proc. ACL*, 2019, pp. 537–546.
- [45] S. Wu, H. Fei, Y. Ren, D. Ji, and J. Li, "Learn from syntax: Improving pair-wise aspect and opinion terms extraction with rich syntactic knowledge," in *Proc. 30th Int. Joint Conf. Artif. Intell. (IJCAI)*, Z. Zhou, Ed., 2021, pp. 3957–3963.
- [46] M. Mitchell, J. Aguilar, T. Wilson, and B. Van Durme, "Open domain targeted sentiment," in *Proc. EMNLP*, 2013, pp. 1643–1654.
- [47] Z. Toh and W. Wang, "DLIREC: Aspect term extraction and term polarity classification system," in *Proc. SemEval*, 2014, pp. 235–240.
- [48] Y. Yin, F. Wei, L. Dong, K. Xu, M. Zhang, and M. Zhou, "Unsupervised word and dependency path embeddings for aspect term extraction," in *Proc. IJCAI*, 2016, pp. 2979–2985.
- [49] X. Li and W. Lam, "Deep multi-task learning for aspect term extraction with memory interaction," in *Proc. EMNLP*, 2017, pp. 2886–2892.
- [50] X. Li, L. Bing, P. Li, W. Lam, and Z. Yang, "Aspect term extraction with history attention and selective transformation," in *Proc. IJCAI*, 2018, pp. 4194–4200.
- [51] G. Berend, "Opinion expression mining by exploiting keyphrase extraction," in *Proc. IJNLP*, 2011, pp. 1162–1170.
- [52] K. Liu, L. Xu, and J. Zhao, "Opinion target extraction using word-based translation model," in *Proc. EMNLP*, 2012, pp. 1346–1356.
- [53] B. Yang and C. Cardie, "Joint inference for fine-grained opinion extraction," in *Proc. ACL*, 2013, pp. 1640–1649.
- [54] A. Katiyar and C. Cardie, "Investigating LSTMs for joint extraction of opinion entities and relations," in *Proc. ACL*, 2016, pp. 919–929.
- [55] S. Chen, J. Liu, Y. Wang, W. Zhang, and Z. Chi, "Synchronous double-channel recurrent network for aspect-opinion pair extraction," in *Proc. ACL*, 2020, pp. 6515–6524.
- [56] S. Chen, Y. Wang, J. Liu, and Y. Wang, "Bidirectional machine reading comprehension for aspect sentiment triplet extraction," in *Proc. AAAI*, 2021, pp. 12666–12674.
- [57] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," in *Proc. ICLR*, 2015, pp. 1–15.
- [58] T. Luong, H. Pham, and C. D. Manning, "Effective approaches to attention-based neural machine translation," in *Proc. EMNLP*, 2015, pp. 1412–1421.
- [59] Q. Wang *et al.*, "Learning deep transformer models for machine translation," in *Proc. ACL*, 2019, pp. 1810–1822.
- [60] A. M. Rush, S. Chopra, and J. Weston, "A neural attention model for abstractive sentence summarization," in *Proc. EMNLP*, 2015, pp. 379–389.
- [61] S. Chopra, M. Auli, and A. M. Rush, "Abstractive sentence summarization with attentive recurrent neural networks," in *Proc. NAACL*, 2016, pp. 93–98.
- [62] X. Zhang, F. Wei, and M. Zhou, "HIBERT: Document level pre-training of hierarchical bidirectional transformers for document summarization," in *Proc. ACL*, 2019, pp. 5059–5069.
- [63] I. V. Serban, A. Sordani, Y. Bengio, A. C. Courville, and J. Pineau, "Building end-to-end dialogue systems using generative hierarchical neural network models," in *Proc. AAAI*, 2016, pp. 3776–3784.
- [64] I. V. Serban *et al.*, "A hierarchical latent variable encoder-decoder model for generating dialogues," in *Proc. AAAI*, 2017, pp. 3295–3301.
- [65] E. Ekstedt and G. Skantze, "TurnGPT: A transformer-based language model for predicting turn-taking in spoken dialog," in *Proc. Findings Assoc. Comput. Linguistics, EMNLP*, 2020, pp. 2981–2990.
- [66] X. Ma, Z. Hu, J. Liu, N. Peng, G. Neubig, and E. Hovy, "Stack-pointer networks for dependency parsing," in *Proc. ACL*, 2018, pp. 1403–1414.
- [67] H. Fei, M. Zhang, B. Li, and D. Ji, "End-to-end semantic role labeling with neural transition-based model," in *Proc. AAAI Conf. Artif. Intell.*, 2021, pp. 566–575.
- [68] D. Fernández-González and C. Gómez-Rodríguez, "Transition-based semantic dependency parsing with pointer networks," in *Proc. ACL*, 2020, pp. 7035–7046.
- [69] O. Vinyals, M. Fortunato, and N. Jaitly, "Pointer networks," in *Proc. NIPS*, 2015, pp. 2692–2700.
- [70] H. Fei, F. Li, B. Li, and D. Ji, "Encoder-decoder based unified semantic role labeling with label-aware syntax," in *Proc. AAAI Conf. Artif. Intell.*, 2021, pp. 1479–1488.
- [71] H. Fei, D. Ji, B. Li, Y. Liu, Y. Ren, and F. Li, "Rethinking boundaries: End-to-end recognition of discontinuous mentions with pointer networks," in *Proc. AAAI Conf. Artif. Intell.*, 2021, pp. 12785–12793.
- [72] J. Libovický and J. Helcl, "End-to-end non-autoregressive neural machine translation with connectionist temporal classification," in *Proc. EMNLP*, 2018, pp. 3016–3021.
- [73] X. Ma, C. Zhou, X. Li, G. Neubig, and E. Hovy, "FlowSeq: Non-autoregressive conditional sequence generation with generative flow," in *Proc. EMNLP*, 2019, pp. 4282–4292.
- [74] J. Guo, L. Xu, and E. Chen, "Jointly masked sequence-to-sequence model for non-autoregressive neural machine translation," in *Proc. ACL*, 2020, pp. 376–385.
- [75] P. Bojanowski, E. Grave, A. Joulin, and T. Mikolov, "Enriching word vectors with subword information," *Trans. Assoc. Comput. Linguistics*, vol. 5, pp. 135–146, Dec. 2017.
- [76] H. W. Kuhn, "The Hungarian method for the assignment problem," *Naval Res. Logistics Quart.*, vol. 2, nos. 1–2, pp. 83–97, Mar. 1955.
- [77] J. Pennington, R. Socher, and C. Manning, "Glove: Global vectors for word representation," in *Proc. EMNLP*, 2014, pp. 1532–1543.
- [78] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in *Proc. NAACL*, 2019, pp. 4171–4186.
- [79] H. Fei, Y. Ren, Y. Zhang, D. Ji, and X. Liang, "Enriching contextualized language model from knowledge graph for biomedical information extraction," *Briefings Bioinf.*, vol. 22, no. 3, May 2021, Art. no. bbaa110.
- [80] M. Peters *et al.*, "Deep contextualized word representations," in *Proc. NAACL*, 2018, pp. 2227–2237.
- [81] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," in *Proc. 13th Int. Conf. Artif. Intell. Statist.*, 2010, pp. 249–256.



Hao Fei received the B.E. degree from Xidian University, Xi'an, China, in 2016, and the M.E. degree from Wuhan University, Wuhan, China, in 2018, where he is currently pursuing the Ph.D. degree with the School of Cyber Science and Engineering.

He has been working on natural language processing (NLP) and data mining. He has published related papers in related journals and conferences, such as the Annual Meeting of the Association (ACL), the AAAI Conference on Artificial Intelligence (AAAI), the International Joint Conference on Artificial Intelligence (IJCAI), the Web Conference (WWW), the Conference on Empirical Methods in NLP (EMNLP), the International Conference on Computational Linguistics (COLING), IEEE TRANSACTIONS ON NEURAL NETWORKS AND LEARNING SYSTEMS, and IEEE/ACM TRANSACTIONS ON AUDIO, SPEECH AND LANGUAGE PROCESSING (TASLP). His current research interests include syntactic and semantic parsing, sentiment analysis, and information extraction.



Yue Zhang (Member, IEEE) received the B.E. degree in computer science from Tsinghua University, Beijing, China, in 2003, and the M.S. and Ph.D. degrees from Oxford University, Oxford, U.K., in 2006 and 2009, respectively.

He was an Assistant Professor with the Singapore University of Technology and Design, Singapore, from 2012 to 2018. He is currently an Associate Professor with Westlake University, Hangzhou, China. His research interests include natural language processing (NLP), text mining, and machine learning.



Yafeng Ren received the Ph.D. degree from Wuhan University, Wuhan, China, in 2015.

He was a Post-Doctoral Research Fellow with the Singapore University of Technology and Design, Singapore, from 2015 to 2016. He is currently an Associate Professor with the Guangdong University of Foreign Studies, Guangzhou, China. He has been working on natural language processing (NLP) over the past ten years. He has published 20 related papers in journals and conferences, including the AAAI Conference on Artificial Intelligence (AAAI), the Conference on Empirical Methods in NLP (EMNLP), and the International Conference on Computational Linguistics (COLING). His research interests include opinion mining, biomedical text mining, and bioinformatics.



Donghong Ji received the B.E., M.E., and Ph.D. degrees from the School of Computer Science, Wuhan University, Wuhan, China, in 1988, 1991, and 1995, respectively.

He was a Post-Doctoral Research Fellow with Tsinghua University, Beijing, China, from 1995 to 1998. From 1998 to 2008, he was a Researcher with the Institute for Infocomm Research, Singapore. He is currently a Professor with Wuhan University. His research interests include natural language processing (NLP), machine learning, and data mining.