Applied Intelligence
https://doi.org/10.1007/s10489-022-03286-w

Synchronously tracking entities and relations in a syntax-aware parallel architecture for aspect-opinion pair extraction

Yue Zhang[1,3] · Tao Peng[1,2,3] · Ridong Han[1,3] · Jiayu Han[4] · Lin Yue[5] · Lu Liu[1,2,3]

Accepted: 22 January 2022

Abstract

Aspect-Opinion Pair Extraction (AOPE) task aims to capture each aspect with its corresponding opinions in user reviews. Entity recognition and relation detection are two fundamental subtasks of AOPE. Although recent works take interaction into account, the two subtasks are still relatively independent during calculation. Furthermore, since AOPE task has not been formally proposed for a long time, syntactic information does not attract much attention in the current deep learning models for AOPE. In this paper, we propose a model for **S**ynchronously **T**racking **E**ntities and **R**elations (STER) to deal with AOPE. Specifically, we design a network consisting of a bank of gated RNNs, where we can track all entities of a review sentence in parallel. STER utilizes three features, i.e., context, syntax and relation, to learn the representation of each tracked entity and calculate the correlated degree between all entities synchronously at each time step. The entity representation and the correlated degree are highly dependent during calculation. Finally, they will be used for entity recognition and relation detection, respectively. Therefore, in STER, the two subtasks of AOPE can achieve sufficient interaction, which enhances their mutual heuristic effect heavily. To verify the effectiveness and adaptiveness of our model, we conduct experiments on two annotation versions of SemEval datasets. The results demonstrate that STER not only achieves advanced performances but adapts to different annotation strategies well.

Keywords Sentiment analysis · Entity recognition · Relation detection · Syntax

1 Introduction

Review resources on the web can reflect the quality of a product or service based on the sentiment polarity of reviewers, and researches on sentiment analysis of these reviews have high application values for business and society as a whole [1–4]. However, the sentiment contained in a review can be mixed. For example, in the review "*This laptop is good in performance but poor in appearance,*" we cannot easily judge the sentiment polarity at the sentence level because the reviewer has different opinions for different aspects. Aspect Based Sentiment Analysis (ABSA) is a fine-grained sentiment analysis task, which analyzes user's sentiment tendencies towards various aspects of a product or service in a review. Aspect extraction [5] and sentiment polarity classification for the extracted aspects [6] are two primary subtasks of ABSA. Aspect extraction is a unique and essential task for ABSA compared to document-level and sentence-level sentiment analyses, and numerous scholars have proposed a variety of approaches to solve this task [7–10]. In addition, to support both subtasks of ABSA and to make users capture the advantages and disadvantages of various aspects in a short time, scholars have also researched the opinion extraction task [11–13]. Later, based on the correlations of aspects and opinions, numerous models have been presented for their co-extraction. Some researchers exploit multi-task learning frameworks to co-extract aspects and opinions in a joint rather than pipeline manner to avoid error propagation and improve co-extraction performances [14–17]. Although their works have considered the relations between aspects and opinions, most of their methods used for relation detection are relatively simple and have trouble detecting complex correlations. Furthermore, they usually ignore the interaction between relation detection and entity recognition, which restricts the mutual heuristic effect. For example, an entity identified as an aspect or opinion term is more likely to correlate with other entities; correspondingly, an entity with a higher correlated degree with others

✉ Tao Peng
tpeng@jlu.edu.cn

Extended author information available on the last page of the article.

Published online: 12 March 2022

is more likely to be an aspect or opinion term, whereas most co-extraction studies only consider the latter.

Additionally, these co-extraction studies do not extract aspects and opinions in pairs. As the instance shown in Fig. 1, co-extraction only recognizes the role of entities, while pair-wise extraction intuitively reflects the sentiment for each aspect. Hence Aspect-Opinion Pair Extraction (AOPE) has been proposed recently [18–20]. For AOPE, entity recognition and relation detection are two fundamental subtasks. The relation detection unit, which interacts with the entity recognition unit, can match more relations between aspects and opinions, such as one-to-many and many-to-one. Correspondingly, the performance of the entity recognition unit can also be improved with these relations. Nevertheless, during our study, we found that the sufficiency of interaction in most AOPE models still has room for improvement, where the two subtasks remain relatively independent during calculation and the number of interactions is few. Furthermore, since AOPE task has not been formally proposed for a long time, syntactic information does not attract the attention of many scholars in the current deep learning models for AOPE. The roles and relations of aspects and opinions usually appear regularity in syntax, which can provide more clues for both subtasks of AOPE. Some AOPE models that take syntactic information into account still rely on external information, such as syntactic dependency trees and labels, and they require expensive effort for parsing and highly depend on the parsing performances [21, 22]. More importantly, there are many web reviews with non-standard grammar. In this case, the performance of the model relying on external knowledge is likely to be limited. To sum up, AOPE is still an under-investigated task.

In this paper, we propose an end-to-end joint learning model for **S**ynchronously **T**racking **E**ntities and **R**elations (STER) to deal with AOPE in a syntax-aware parallel architecture. First, based on the excellent performance of Bidirectional Encoder Representations from Transformers (BERT) [23] in representation learning, we adopts it to learn the initial representation of each token in a review sentence. Then, we design a neural network consisting of a bank of gated RNNs, with which we can track all tokens in parallel, synchronously update the representations of all tracked tokens and calculate the correlated degree between the tracked and input tokens utilizing three features of context, syntax and relation at each time step. We use the dimensional hierarchy of vectors to store syntactic information in sentences without any external resources. All the learned information comes from the review sentences. More importantly, the token representation and the correlated degree are highly dependent during calculation. In the end, we feed the final token representations into Conditional Random Field (CRF) [24] for entity labeling and use the correlated degree for relation classification. Thereupon, the two subtasks of AOPE can achieve sufficient interaction in STER. To verify the effectiveness and adaptiveness of STER, we conduct a serial of experiments on the datasets based on SemEval benchmarks with two different annotation versions. The results show that STER achieves good results on both annotation versions.

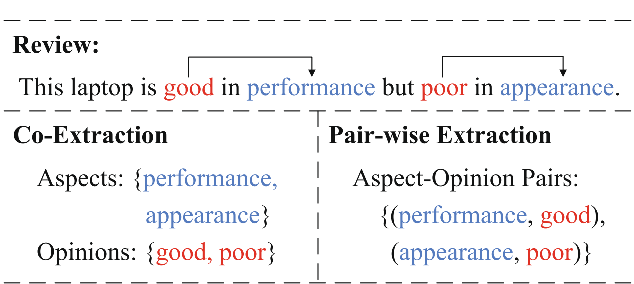


Fig. 1 An example of the comparison between co-extraction and pair-wise extraction

In summary, the main contributions of our work are concluded as follows:

1. We propose an end-to-end joint learning model, STER. By designing the parallel architecture, STER can track entities and relations synchronously and exploit rich information, including context, syntax and relation, to assist in both subtasks of AOPE. All the learned information comes from the review sentences without external resources.
2. In STER, entity recognition and relation detection can achieve sufficient interaction with the high dependence of calculation, which heavily enhances their mutual heuristic effect.
3. The framework of STER provides a general solution for AOPE and AOPE-like tasks, i.e., updating the entity representation and calculating the correlated degree of two entities simultaneously on each unit of an $N \times N$ grid. The algorithms inside the grid are easily modified and replaced, so STER has strong flexibility and scalability.
4. We conduct experiments on two annotation versions of datasets based on SemEval benchmarks. The results verify that our model achieves advanced performances and adapts to different annotation strategies well.

2 Related work

Early works on aspect and opinion extraction tasks are rule-based. Hu and Liu [25] considered frequent nouns or noun phrases in the review text as opinion targets based on association rules, while judging whether infrequent nouns or noun phrases were opinion targets based on their distances from the opinion words. Popescu and Etzioni [26] used Point-wise Mutual Information (PMI) to filter these infrequent nouns or noun phrases, so as to retain the true opinion targets. Zhuang et al. [27] focused on the dependency relation and developed the dependency template based on the high frequency of dependencies between opinion targets and opinion words for the extraction tasks of movie reviews. Qiu et al. [28] used a domain corpus to find domain-related opinion words, and the noun opinion targets could be extracted with the extracted opinion words and the syntactic dependency trees. The performances of these models rely heavily on the used rules and easily suffer from limitations because the patterns not in the rules cannot be detected. Traditional machine learning methods usually treat the extraction tasks as a sequence labeling problem. Among various sequence labeling models, CRF is widely used due to its powerful performance. Jakob and Gurevych [29] first used CRF in the opinion target extraction task and exploited several features, including token, part-of-speech (POS), short dependency path, word distance, and opinion sentence. In addition to the features of linear chain structure, Li et al. [30] also considered the conjunction structure and the syntactic tree structure, i.e., the linguistic structure, as the features for CRF to extract both opinion targets and opinion words. However, these CRF-based extraction models generally depend on hand-crafted features that are used in linear combinations rather than high-level interactions.

To automatically capture and more effectively combine features, neural networks have been applied to many aspect and opinion extraction studies. Poria et al. [31] were the first to present a deep learning method for the aspect extraction task and exploited a 7-layer deep convolutional neural network to capture features for each word in a sentence. In addition, based on the effectiveness of the syntactic structure for the opinion target extraction task, many scholars have used neural network to extract dependency features from dependency trees. Luo et al. [9] further considered the propagation direction of the dependency structure and proposed a bidirectional dependency tree network to obtain two representations from the bottom-up and top-down propagation on a dependency tree. Finally, they combined both tree-structured and sequential features to deal with the aspect extraction task. Fan et al. [11] designed a target-fused sequence labeling neural network to perform the opinion extraction task. Specifically, in their network, the target information is transferred to the left and right contexts in opposite directions with two LSTMs. Then the left and right contexts are combined with the global context which obtained through a bidirectional LSTM to gain the representation of each word, and finally the sequence labeling is performed. In addition to saving time in constructing numerous manual features, the extraction performances of these networks are generally better than those CRF-based models performed with hand-crafted features. However, the above researches just studied one of the extraction tasks or ignored the interaction between the two extraction tasks. To complete the whole ABSA task, scholars need to extract aspects firstly, then detect opinions for each extracted aspect, and finally, judge the sentiment polarity for each aspect according to its opinions. Obviously, this pipeline method would suffer from error propagation.

Afterwards, joint learning methods for co-extraction are proposed. Scholars have presented various methods to detect the relations between aspects and opinions, which also assists in recognizing the roles of entities. Apart from the label system for entity recognition, Katiyar et al. [14] also built a label system for relation detection based on the distance between entities to jointly extract the relations. In the two works of Wang et al. [32, 33], they respectively exploited RNN and attention mechanism for encoding the aspect-opinion relations in high-level representation learning based on the dependency parse tree. Yu et al. [16] designed a multi-task learning framework to implicitly capture the aspect-opinion relations and proposed a global inference method to explicitly model syntactic constraints among the two extraction tasks. Due to the limitation of the dependency path template, Dai and Song [17] proposed automatic mining rules of dependency relation, which could capture relations more flexibly. However, though these works exploit the relations between aspects and opinions to perform co-extraction, they are still limited. First, some methods for relation detection only depend on the distance or syntax, which cannot ensure their adaptability to different corpora or annotation strategies, especially for some online reviews whose language structures lack standardization. Second, these studies generally utilize relation detection to assist in entity recognition but ignore the interaction between them, which limits the two processes to be mutually beneficial [34]. Third, compared with co-extraction, pair-wise extraction is more conducive to ABSA and its downstream tasks. Therefore, AOPE task is proposed.

AOPE is also available in both pipeline and joint learning methods, and the difference is whether relation detection is performed after or simultaneously with entity recognition. For avoiding error propagation, the latter is adopted by the majority. Chen et al. [18] set up the units of entity recognition and relation detection as two channels. To achieve interaction, they built a synchronization unit to influence updating the input of the next execution for both channels. Their experimental results demonstrate the effectiveness of interaction between the two processes. However, in terms of structure, the connection between the two channels is not close enough, and the number of interactions is few. Zhao et al. [19] combined tokens as spans in different lengths by enumeration and designed two encoders to learn the representation of each span. Finally, all candidate spans were used to perform entity recognition and relation detection simultaneously. In this model, the connection between the two subtasks of AOPE is only sharing the span representations. Although these scholars have taken the interaction into account, the entity recognition and relation detection subtasks are still relatively independent during calculation in their works. Gao et al. [35] took an alternative approach to deal with AOPE. First, they extracted the aspects based on a span-wise scheme and constructed the questions about the extracted aspects. Then, they treated the corresponding opinion term extraction task as the reading comprehension task. Finally, they combined the two processes for joint learning and obtained good results. However, none of the above three studies consider syntactic information, which has been frequently needed in previous aspect and opinion extraction studies and can provide more clues for both subtasks of AOPE. In the two works of Wu et al. [21, 22], they exploited GCN to model the dependency edges and labels for better span boundary and relation detections. They have also achieved good results on AOPE, but like most ABSA models that utilize syntactic parsing information, their two models crucially depend on the grammatical accuracy of the review sentences and the performance of the parsing algorithm. In this paper, we further explore AOPE task and propose an end-to-end model, STER, which can achieve sufficient interaction of subtasks and learn syntactic information without the external resources.

3 Model

3.1 Task description

Given a token sequence $S = \{s_1, s_2, ..., s_N\}$ of a review sentence, AOPE aims to extract the set of aspect-opinion pairs $AOP = \{(a_k, o_l), ...\}$, where a_k and o_l consisting of one or more tokens from S represent an aspect term and its corresponding opinion term, respectively. As mentioned above, AOPE is generally divided into entity recognition and relation detection subtasks. Specifically, entity recognition is responsible for assigning a label y_i^E to each token s_i, and $y_i^E \in \{B-A, I-A, B-P, I-P, O\}$ where B/I means that the token is the start/inside of an aspect (A) or opinion (P) term while O denotes that the token does not belong to any aspect or opinion term. Relation detection is responsible for assigning a label $y_{m,n}^R$ to each token pair (s_m, s_n), and $y_{m,n}^R \in \{1, 0\}$ where 1 indicates that there is a pair-wise relation between s_m and s_n while 0 is opposite.

3.2 Framework

To deal with the above tasks, we design a model for **S**ynchronously **T**racking **E**ntities and **R**elations (STER). The framework of STER is shown in Fig. 2. We utilize BERT to encode the input sentence for learning the initial context representation of each token. In order to make entity recognition and relation detection interact, we construct a tracking network through which both subtasks can perform synchronously. This network contains multiple memory cells in parallel to track all tokens in a review, and the

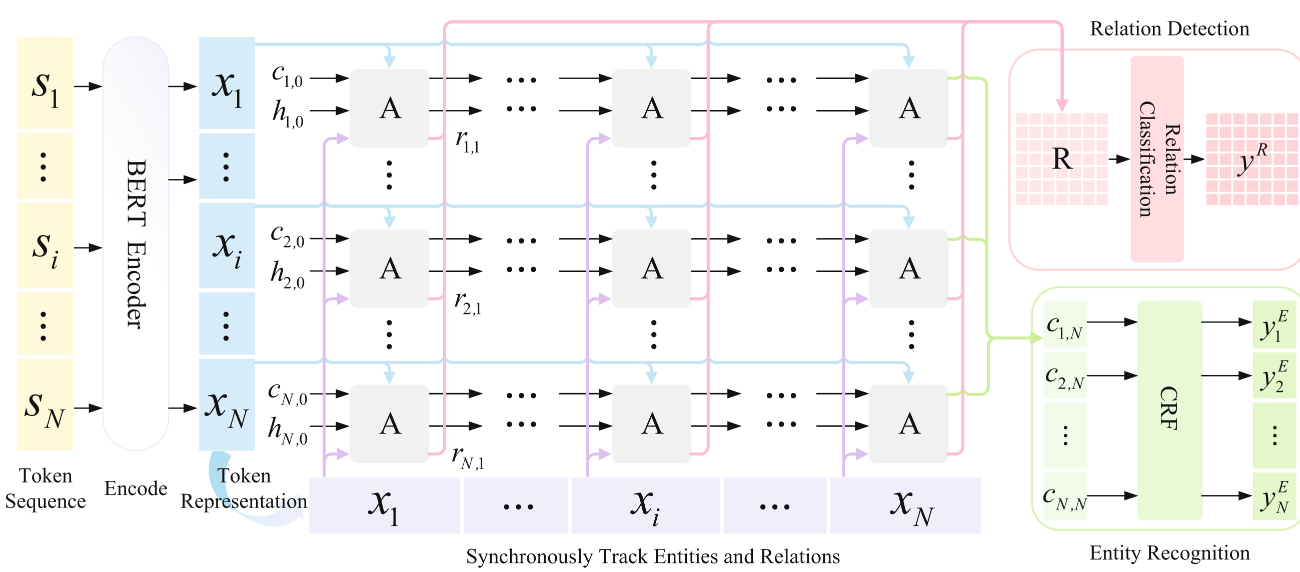


Fig. 2 The framework of STER. We unfold all time steps of each memory cell, and the calculation unit marked with the letter A is internal of a cell, whose details are shown in Fig. 3

input sequence of time steps is also the token sequence. The detailed illustration of a memory cell is provided in Fig. 3. With the help of gates in the cell, all tracked tokens dynamically update their state based on the associated context, syntax and relation as progressing through the input sequence. Moreover, at each time step, according to the representations of tracked token and input token, the network will utilize a relation gate to calculate the probability of two tokens belonging to an aspect-opinion pair, i.e., correlated degree. In a word, the correlated degree influences updating the tracked token representation and is influenced by the token representation at previous time step in turn. Finally, token representations at the last time step and correlated degree at all time steps will be respectively used for sequence labeling with CRF and relation classification. In conclusion, the two subtasks of AOPE can achieve continuous interaction based on the high calculational dependence during the whole entity tracking process. Additionally, this framework can also provide a general solution for AOPE and AOPE-like tasks. The algorithms inside the cell are easily modified and replaced according to the needs of users.

3.3 BERT encoder

Due to the powerful performance of BERT, we adopt it as the encoder to learn the initial representation of each token incorporating contextual information. An input vector e_i of BERT is the sum of three embedding features containing token embedding e_i^t, position embedding e_i^p and segment embedding e_i^s:

$$e_i = e_i^t + e_i^p + e_i^s \tag{1}$$

Token embedding is obtained by tokenizing with WordPiece, i.e., dividing a word into a limited set of public subword units, such as splitting ‘learning’ into ‘learn’ and ‘ing’, which can compromise word validity and character flexibility. Then we respectively add two special symbols [CLS] and [SEP] at the start and the end to obtain the complete token sequence S. And then, the token embedding generates by converting each token s_i to its corresponding id. Position embedding refers to encoding the position information of a token into a feature vector, and segment embedding is used to distinguish different sentences within a paragraph. In the end, the vector sequence $E = \{e_1, e_2, ..., e_N\}$ will be fed into a pre-trained BERT model that leverages Transformer as its main framework. Transformer can capture the bidirectional relation in a sentence and learn rich contextual information. Finally, we can obtain the following context representation of each token:

$$x_i = \mathrm{BERT}(e_i) \tag{2}$$

The sentence sequence $X = \{x_1, x_2, ..., x_i, ..., x_N\}$ will be used to perform the subsequent tasks.

3.4 Tracking network

In this section, we will introduce the tracking network of STER in terms of the parallel architecture, the update strategy of cell state, and network comparison.

3.4.1 Parallel architecture

To deal with AOPE task, we expect to learn token representations combined with rich information for entity recognition and build an $N \times N$ correlation matrix holding the probability of each token pair belonging to an aspect-opinion pair for relation detection. Additionally, it is essential to enhance the mutual heuristic effect between the two subtasks by interacting. Inspired by Recurrent Entity Network (EntNet) [36], we design a neural network of parallel architecture synchronously tracking entities and relations. The

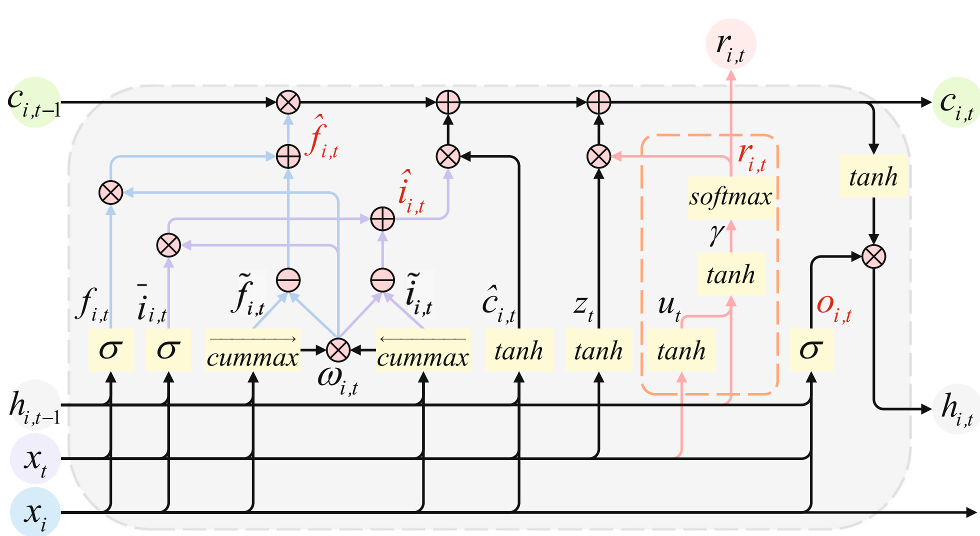


Fig. 3 Illustration of i^{th} memory cell at time step t. The four gates in red font are the final forget, input, output, and relation gates. The calculation of our relation gate is circled with an orange box. ⊗, ⊕, and ⊖ correspond to ∘, + and − in the equations, respectively

original intention of EntNet is tracking the state evolution of specific entities with the story flowing and dealing with question-answering tasks during reading comprehension. When reading a new story fragment, EntNet will calculate the matching degree between the tracked entities and new information, then take it as a gating value to update the state of tracked entities. Thereupon, we associate that if we change the new data from sentence level to token level as Liu et al. [37] do, set the same number of tracked entities as the tokens in a review, and modify the calculation method of matching degree for relation detection, the variant network can calculate the probabilities of all entity pairs being aspect-opinion pairs, based on which the representations of tracked entities are updated. Consequently, we construct a network consisting of a bank of gated RNNs and set all tokens of a review as the tracked targets. The input sequence of the network is also set as the token sequence. The network can update the token representation and calculate the correlated degree synchronously at each time step. Moreover, by designing the update algorithm of cell state, the token representation and the correlated degree are highly dependent during calculation, which takes the interaction of both AOPE subtasks into account.

3.4.2 Cell state update

The cell state update algorithm of a EntNet variant proposed by [37] is shown as follows:

$$\hat{h}_{i,t} = \mathrm{ReLU}(W_h x_t + U_h h_{i,t-1} + V_h k_i) \tag{3}$$
$$\bar{d}_{i,t} = \mathrm{GRU}(\hat{h}_{i,t}, \bar{d}_{i,t-1}) \tag{4}$$
$$g_{i,t} = \sigma(x_t \cdot h_{i,t-1} + x_t \cdot k_i + w_{\bar{d}} \cdot \bar{d}_{i,t}) \tag{5}$$
$$h_{i,t} = h_{i,t-1} + g_{i,t} \odot \hat{h}_{i,t} \tag{6}$$
$$h_{i,t} = \frac{h_{i,t}}{\|h_{i,t}\|} \tag{7}$$

where $\hat{h}_{i,t}$ and $h_{i,t} \in \mathbb{R}^{d_e \times 1}$ are the candidate hidden state and new hidden state of i^{th} tracked entity $k_i \in \mathbb{R}^{d_e \times 1}$ at time step t, respectively. d_e is the dimension of all vectors in the setting of this EntNet variant. $x_t \in \mathbb{R}^{d_e \times 1}$ is the input of time step t. $\bar{d}_{i,t} \in \mathbb{R}^{d_e \times 1}$ is the delayed memory designed for a longer term of memory. $g_{i,t}$ is the update gate, i.e., the matching degree between the tracked entity k_i and the current input x_t. The $\cdot$ symbol is the vector dot product, therefore $g_{i,t}$ is a value. $\odot$ is the Hadamard product. W_h, U_h, $V_h \in \mathbb{R}^{d_e \times d_e}$, and $w_{\bar{d}} \in \mathbb{R}^{d_e \times 1}$ are trainable weight matrices and vector shared by all memory cells at each time step. However, due to the characteristics of AOPE, the update strategy of cell state in STER is different from EntNet.

Syntactic information can provide clues for AOPE. We can exploit the hierarchical level of an entity and the dependency relation between entities to assist in both subtasks of AOPE. In previous works, syntactic information is often introduced with external dependency path templates or syntactic parsers, which causes a limitation. In 2019, Shen et al. [38] proposed Ordered Neurons LSTM (ON-LSTM), integrating the hierarchy of syntactic structure into LSTM. More importantly, both EntNet and ON-LSTM are RNN structures, which makes it possible to integrate their update algorithm of cell state. In addition, we can obtain long-term memory without a separate unit of delayed memory due to the characteristics of LSTM. The cell state update algorithm of ON-LSTM is shown as follows:

$$f_t = \sigma(W_f x_t + U_f h_{t-1} + b_f) \tag{8}$$
$$i_t = \sigma(W_i x_t + U_i h_{t-1} + b_i) \tag{9}$$
$$o_t = \sigma(W_o x_t + U_o h_{t-1} + b_o) \tag{10}$$
$$\hat{c}_t = \tanh(W_c x_t + U_c h_{t-1} + b_c) \tag{11}$$
$$\tilde{f}_t = \mathrm{cummax}(W_{\tilde{f}} x_t + U_{\tilde{f}} h_{t-1} + b_{\tilde{f}}) \tag{12}$$
$$\tilde{i}_t = 1 - \mathrm{cummax}(W_{\tilde{i}} x_t + U_{\tilde{i}} h_{t-1} + b_{\tilde{i}}) \tag{13}$$
$$\omega_t = \tilde{f}_t \circ \tilde{i}_t \tag{14}$$
$$\hat{f}_t = f_t \circ \omega_t + (\tilde{f}_t - \omega_t) \tag{15}$$
$$\hat{i}_t = i_t \circ \omega_t + (\tilde{i}_t - \omega_t) \tag{16}$$
$$c_t = \hat{f}_t \circ c_{t-1} + \hat{i}_t \circ \hat{c}_t \tag{17}$$
$$h_t = o_t \circ \tanh(c_t) \tag{18}$$

where f_t, i_t, and $o_t \in \mathbb{R}^{d_h \times 1}$, are the fundamental forget, input and output gates, respectively. d_h is the dimension of hidden state. $\hat{c}_t$ and $h_t \in \mathbb{R}^{d_h \times 1}$ are the candidate cell state and the hidden state at time step t, respectively. Their equations are the same as the standard LSTM. $\mathbf{W} \in \mathbb{R}^{d_h \times d_i}$, $\mathbf{U} \in \mathbb{R}^{d_h \times d_h}$ and $\mathbf{b} \in \mathbb{R}^{d_h \times 1}$ are the weight matrices and bias for gates. d_i is the dimension of the input vector x_t. ON-LSTM reflects hierarchical information into the vector of cell state. It divides the dimensions of a representation vector into intervals. Moreover, different intervals represent different syntactic levels with varying strategies of update. Specifically, ON-LSTM sets $\tilde{f}_t$ and $\tilde{i}_t \in \mathbb{R}^{d_h \times 1}$ to respectively split the intervals of high level and low level in the cell state with activation function $\mathrm{cummax}(...) = \mathrm{cumsum}(\mathrm{softmax}(...))$. $\omega_t \in \mathbb{R}^{d_h \times 1}$ denotes their overlapping interval. The $\circ$ symbol also represents the Hadamard product. According to the characteristics of syntax, for high hierarchy, the corresponding dimension interval should keep historical information for a long time, while for low hierarchy, it should update with a high frequency. As for their overlapping interval, both forgetting and updating information are needed, so ON-LSTM updates this interval as the standard LSTM does. In the end, we can get the final forget gate $\hat{f}_t$ and the final input gate $\hat{i}_t$ to calculate the cell state c_t and the hidden state h_t at time step t.

A marked difference between EntNet and ON-LSTM is that EntNet updates the cell state aiming at a certain entity, while ON-LSTM updates aiming at the entire sentence. EntNet learns information related to the tracked entity, and its equations all involve the tracked entity k_i. Thereupon, in order to inherit the parallel architecture of EntNet, we perform entity extension to ON-LSTM by involving the representation of tracked token x_i in the calculations of various gates and the candidate cell state as follows:

$$f_{i,t} = \sigma(W_f x_t + U_f h_{i,t-1} + V_f x_i + b_f) \tag{19}$$

$$\bar{i}_{i,t} = \sigma(W_{\bar{i}} x_t + U_{\bar{i}} h_{i,t-1} + V_{\bar{i}} x_i + b_{\bar{i}}) \tag{20}$$

$$o_{i,t} = \sigma(W_o x_t + U_o h_{i,t-1} + V_o x_i + b_o) \tag{21}$$

$$\hat{c}_{i,t} = \tanh(W_c x_t + U_c h_{i,t-1} + V_c x_i + b_c) \tag{22}$$

$$\tilde{f}_{i,t} = \text{cummax}(W_{\tilde{f}} x_t + U_{\tilde{f}} h_{i,t-1} + V_{\tilde{f}} x_i + b_{\tilde{f}}) \tag{23}$$

$$\tilde{i}_{i,t} = 1 - \text{cummax}(W_{\tilde{i}} x_t + U_{\tilde{i}} h_{i,t-1} + V_{\tilde{i}} x_i + b_{\tilde{i}}) \tag{24}$$

where x_i is the same role as k_i in EntNet and $1 \le i \le N$, which denotes that we set N parallel tracking chains.[1] x_t is the current input token and $1 \le t \le N$, which means the number of time steps is also N. $\boldsymbol{W} \in \mathbb{R}^{d_h \times d_i}$, $\boldsymbol{U} \in \mathbb{R}^{d_h \times d_h}$, $\boldsymbol{V} \in \mathbb{R}^{d_h \times d_i}$ and $\boldsymbol{b} \in \mathbb{R}^{d_h \times 1}$ are the weight matrices and bias for gates, where d_h and d_i are the dimension of hidden state and input token, respectively. Additionally, we add a bar on the top of original input gate symbol to avoid confusion of symbols.

Naturally, the equations of the final forget and input gates change accordingly:

$$\omega_{i,t} = \tilde{f}_{i,t} \circ \tilde{i}_{i,t} \tag{25}$$

$$\hat{f}_{i,t} = f_{i,t} \circ \omega_{i,t} + (\tilde{f}_{i,t} - \omega_{i,t}) \tag{26}$$

$$\hat{i}_{i,t} = \bar{i}_{i,t} \circ \omega_{i,t} + (\tilde{i}_{i,t} - \omega_{i,t}) \tag{27}$$

With the above gates, our tracking network can learn the contextual and syntactic information related to each tracked token when performing cell state update.

For relation detection, we design a relation gate whose value is the correlated degree between the tracked token and the input token. Its function is similar to $g_{i,t}$, which denotes the matching degree in EntNet. However, (5) is not suitable for calculating the correlated degree in AOPE. The similarity obtained by dot product indicates the distance in vector space where an aspect and its opinion are generally not close. Motivated by [18], we use the following equations to calculate the correlated degree:

$$u_t = \tanh(W_u x_t) \tag{28}$$

$$\gamma(u_t, h_{i,t-1}) = W_r^1 \tanh(W_r^2 u_t + W_r^3 h_{i,t-1}) \tag{29}$$

where $h_{i,t-1}$ can be seen as the latest representation of the tracked token x_i. In order to map x_t and $h_{i,t-1}$ to the same vector space, we perform a linear transformation on x_t, where $W_u \in \mathbb{R}^{d_h \times d_i}$ is the transformation matrix. Referring to the calculation of the hidden state, i.e. (18), we also adopt tanh function to activate x_t. Then, we can get u_t and feed it with $h_{i,t-1}$ into the scoring function γ to calculate their correlated degree. The matrices W_r^2, $W_r^3 \in \mathbb{R}^{d_h \times d_h}$, and $W_r^1 \in \mathbb{R}^{1 \times d_h}$ are trainable parameters.

With the softmax function, we can obtain the probability distribution of correlated degree between current input token and all tracked tokens:

$$r_{i,t} = \frac{\exp(\gamma(u_t, h_{i,t-1}))}{\sum_{j=1}^{N} \exp(\gamma(u_t, h_{j,t-1}))} \tag{30}$$

We also set $r_{i,t}$ as the relation gate and update the new cell state as follows:

$$z_t = \tanh(W_z x_t) \tag{31}$$

$$c_{i,t} = \hat{f}_{i,t} \circ c_{i,t-1} + \hat{i}_{i,t} \circ \hat{c}_{i,t} + r_{i,t} \circ z_t \tag{32}$$

where $r_{i,t} \circ z_t$ indicates the relational information between x_i an x_t. Consistent with (28), we also perform a linear transformation and a tanh activation on x_t, where $W_z \in \mathbb{R}^{d_h \times d_i}$. To sum up, during update, forget gate $\hat{f}_{i,t}$ and input gate $\hat{i}_{i,t}$ are responsible for learning contextual and syntactic information, while relation gate $r_{i,t}$ is responsible for learning relational information.

In the end, we can obtain the new hidden state of i^{th} tracked token x_i at time step t as follows:

$$h_{i,t} = o_{i,t} \circ \tanh(c_{i,t}) \tag{33}$$

According to (30), (32) and (33), it can be observed that the hidden state and the relation gate are highly dependent during calculation. Moreover, cell state[2] at the last time step and correlated degree at all time steps will be used for sequence labelling and relation classification, respectively. Thereupon, entity recognition and relation detection can achieve sufficient interaction in STER.

3.4.3 Network comparison

Although we design the above tracking network inspired by EntNet and ON-LSTM, there are still many differences.

Compare with EntNet The idea of tracking entities comes from EntNet, but the structure of our network is not identical to it. First, the number of tracked entities is no more than 20 in EntNet but is no more than 100 in our network. We switch the input of each time step from sentence level to token level as Liu et al. [37] do, and the calculational complexity at each time step would decreases. Correspondingly, the number of parallel cells is able to increase. In addition, while the granularity of input becomes smaller, the term of memory in the network should be longer, and Liu et al. [37] set a delayed memory unit in their EntNet variants. However,

[1] In our datasets, the length of a review sentence does not exceed 100 tokens. As for longer sentences, slicing should be adopted.

[2] We take cell states instead of hidden states as the final token representations, which can obtain better results.

in our network, due to the characteristics of LSTM, we can obtain long-term memory without setting up a separate delayed memory unit. More importantly, apart from parallel architecture, our update strategy of cell state, which aims at AOPE task, is entirely different from EntNet.

Compare with ON-LSTM Our cell state update strategy is designed with reference to ON-LSTM but is more targeted. Specifically, ON-LSTM learns information based on the whole sentence, while our network learns information based on the tracked token. Moreover, compared with ON-LSTM, our network also learns relational information in addition to contextual and syntactic information. Apart from gates in ON-LSTM, we design a relation gate to calculate the correlated degree between the tracked and input tokens. With this gate, we can detect pair-wise relations and exploit such relations to update the representations of all tracked tokens.

3.5 Objective

After passing sequence X through the tracking network, we can obtain the final token representation sequence $C = \{c_{1,N}, c_{2,N}, ..., c_{N,N}\}$ and the correlation matrix $R \in \mathbb{R}^{N\times N}$ whose element is $r_{i,t}$. Then we respectively perform sequence labelling and relation classification with them.

Entity Recognition Compared with classifiers, CRF considers the correlation with adjacent labels and searches the optimal global solution by calculating the joint probability distribution of the entire sequence, which avoids labelling bias. Hence, we take CRF as our sequence labelling model for entity recognition. Formally, CRF calculates the probability of a label sequence $y^E = \{y_1^E, y_2^E, ..., y_N^E\}$ as follows:

$$P(y^E|C) = \frac{\exp(\alpha(C, y^E))}{\sum_{y^{E'}\in Y^E} \exp(\alpha(C, y^{E'}))} \tag{34}$$

where Y^E is the set of all possible label sequences for C, and α is the composite score function involving the state score matrix $\tilde{S}$ and the transition score matrix $\tilde{T}$ as follows:

$$\alpha(C, y^E) = \sum_{i=1}^{N}(\tilde{S}_{i,y_i^E} + \tilde{T}_{y_{i-1}^E, y_i^E}) \tag{35}$$

$$\tilde{S}_i = W_{\tilde{s}} c_{i,N} + b_{\tilde{s}} \tag{36}$$

where $\tilde{S}_{i,y_i^E}$ denotes the score of $c_{i,N}$ labeled as y_i^E based on its own feature, and $\tilde{T}_{y_{i-1}^E, y_i^E}$ measures the transition score from y_{i-1}^E to y_i^E. The matrices $W_{\tilde{s}} \in \mathbb{R}^{5\times d_h}$ and $b_{\tilde{s}} \in \mathbb{R}^{5\times 1}$ are used to create the mapping between $c_{i,N}$ and five labels. Naturally, $\tilde{S}_{i,y_i^E}$ is extracted from $\tilde{S}_i$. In the end, we take the following negative log-likelihood function as the loss of this subtask:

$$L(E) = -\log P(y^{E*}|C) \tag{37}$$

where y^{E*} is the gold label sequence of C.

Relation Detection Formally, relation detection can be seen as a binary classification task identifying a token pair as related or unrelated. With the correlation matrix R, we can obtain the predicted relation distribution $p(y_{m,n}^R|(s_m, s_n))$ of each token pair. The loss of this subtask is generated by calculating the cross-entropy between R and the gold relation matrix $G \in \mathbb{R}^{N\times N}$ as follows:

$$L(R) = -\sum_{m=1}^{N}\sum_{n=1}^{N} p(y_{m,n}^{R*}|(s_m, s_n))\log(p(y_{m,n}^R|(s_m, s_n))) \tag{38}$$

where $p(y_{m,n}^{R*}|(s_m, s_n))$ denotes the gold relation distribution.

Training Objective Finally, by combining the loss of both subtasks, we can get the final training objective of STER:

$$L(\theta) = \lambda_E L(E) + \lambda_R L(R) \tag{39}$$

To balance the two subtasks, we set two hyperparameters λ_E and λ_R as the weights for achieving the best performances of joint learning.

4 Experiments

4.1 Datasets

To evaluate the effectiveness and adaptiveness of STER, we conduct experiments on two sets of datasets from SemEval Challenge Tasks [39–41]. One is provided by [11], and the other is provided by [18]. They annotate aspects and opinions in pairs based on the original SemEval datasets where only the aspect terms are labeled. Moreover, Chen et al. [18] annotates the pair-wise relation following the opinion annotation provided by [32, 33]. Table 1 shows the statistics of their datasets. Note that Chen et al. [18] keeps the reviews without aspect-opinion pairs, so the number of sentences in their datasets is more than [11].

4.2 Experimental settings

For the encoder, we adopt the uncased BERT$_{base}$ model, where the dimension of token representation, i.e., d_i, is 768, and a dropout of 0.5 is applied to the output of the last layer. We use BertAdam as the parameter optimizer, with the fine-tuning learning rate of 2e-5 and the warmup rate of 0.1. The training learning rate of STER is set to 0.001, and after 20 epochs, it will decay as 98% of the original level. The dimension of the hidden state d_h is 300, and 5-dimension is set to a syntactic level. Additionally, a dropout of 0.4 is applied to the linear transformation of the hidden

Table 1 Statistics of datasets

Datasets		[11]				[18]			
		#S	#A	#O	#P	#S	#A	#O	#P
	Train	1259	2064	2098	2356	3041	3693	3512	2809
Res14	Dev	315	487	506	580	–	–	–	–
	Test	493	851	866	1008	800	1134	1014	936
	Train	899	1257	1270	1452	3045	2359	2500	1535
Lap14	Dev	225	332	313	383	–	–	–	–
	Test	332	467	478	547	800	653	677	380
	Train	603	871	966	1038	1315	1205	1217	1231
Res15	Dev	151	205	226	239	–	–	–	–
	Test	325	436	469	493	685	542	516	516
	Train	863	1213	1329	1421	–	–	–	–
Res16	Dev	216	298	331	348	–	–	–	–
	Test	328	456	485	525	–	–	–	–

#S, #A, #O, and #P denote the number of sentences, aspect terms, opinion terms, and aspect-opinion pairs, respectively

state. The maximum sequence length is 100 with a batch size of 5. λ_E and λ_R are respectively set to 1 and 11 to balance subtasks. We determine the above hyper-parameters with cross-validation. In the end, we report the average experimental results by running each model 5 times with random initialization.

4.3 Evaluation metrics

We adopt the frequently-used metric of F1-score to evaluate the performances of models on both subtasks of AOPE, i.e., entity recognition and relation detection. Moreover, entity recognition can be divided into the aspect extraction and opinion extraction subtasks. For entity recognition, an aspect or opinion term may consist of several tokens. Only when all of these tokens are correctly recognized, the aspect or opinion term is considered to be correctly extracted. For relation detection, in addition to the aspect term and its corresponding opinion term are both accurately identified, the average correlated degree of all token pairs belonging to them should exceed a given threshold, then their pair-wise relation is considered to be correctly detected. In our model, the threshold is set to 0.5.

4.4 Baselines

To evaluate STER comprehensively, we select both pipeline and joint learning models to compare.

4.4.1 Pipeline models

Since AOPE has not been proposed for a long time, most ABSA studies just recognize the role of entities instead of extracting aspect-opinion pairs. We select three advanced models of them as half of the pipeline models:

RNCRF [32] A model of recursive neural network for learning the high-level features. The output representations of tokens are fed into CRF for entity recognition.

CMLA [33] A model using multi-layer attentions to recognize the role of each token as well as the relations between them. It co-extracts aspects and opinions with the help of the detected relations.

RINANTE [17] A model that extracts aspects and opinions in a semi-supervised manner. Their proposers present rules for automatic mining of dependent syntax assisting entity recognition.

For AOPE, it is essential to introduce relation detection units based on the results of entity recognition. We also select three advanced models of them as the other half of the pipeline models:

C-GCN [42] An extended graph convolutional networks for relation extraction. It collects information on arbitrary dependency structures and can also be applied to AOPE.

IOG [11] A model of three LSTM-like encoders for extracting opinion words aiming at the targeted aspect. The three encoders learn the information of the targeted aspect, including the left, right, and global contexts.

SDRN-RD [18] The relation detection channel in SDRN, which can calculate the correlated degree with the representations of two arbitrary tokens. SDRN is also our baseline model and will be introduced in detail later.

According to the experiments conducted on SemEval datasets before, we take RINANTE+SDRN-RD [18], CMLA+C-GCN [20] and RINANTE+IOG [20] as our baselines of pipeline methods.

4.4.2 Joint learning models

To prove the advancement of STER, we also select three models that jointly identify entities and relations as our baseline models. All of them are the advanced models at present.

SDRN [18] A synchronous double-channel recurrent network where one channel is responsible for entity recognition, and the other is responsible for relation detection. For interaction, a synchronization mechanism is also set up between them. The best performance of SDRN can be achieved with several cyclic executions.

SpanMlt [19] An end-to-end model that sets the entity granularity to span level instead of token level. After passing BERT or BiLSTM encoder, all candidate spans are used to perform entity recognition and relation detection simultaneously.

GTS [20] A novel tagging scheme, Grid Tagging Scheme, with which model can tag entities and relations simultaneously. To be specific, this scheme adopts a unified grid tagging combining an arbitrary encoder and a designed inference strategy to deal with AOPE.

SDRN has been determined to use the BERT encoder, while SpanMlt and GTS use uncertain encoders. For unified comparison, we all choose BERT encoding for them. In addition, the above pipeline and joint learning models have been experimented on the annotation datasets of [11] or [32, 33] in previous studies, and for convenience and fairness, we mainly compare STER with them on their corresponding annotation datasets.

4.5 Results and analysis

The results of experiments conducted on the annotation datasets of [11] are shown in Table 2. Firstly, the performances of pipeline models are not satisfactory, especially for relation detection. Whether these models recognize entities before detecting relations or extract aspects before identifying the corresponding opinions, they all suffer from error propagation. Furthermore, there is a lack of interaction between their subtasks. On the contrary, joint learning models can utilize the interaction to enhance the mutual heuristic effect and avoid error propagation.

Among these joint learning models, it is evident that STER achieves the best performances for entity recognition and relation extraction subtasks on four datasets, even compared with the advanced models of AOPE. On average, for the four datasets, STER outperforms SpanMlt by 1.82%, 1.57% and 4.67% on AF, OF, and RF, respectively. For GTS, STER outperforms its reproduced model by 2.19%, 2.15% and 1.48% on the three metrics, respectively. For SDRN, STER outperforms its reproduced model by 3.20%, 1.77% and 2.02% on the three metrics, respectively. Based on the above comparison results, we can conclude that SpanMlt outperforms in entity recognition but underperforms in relation detection compared to GTX and SDRN. SpanMlt handles an entity at a larger granularity, i.e., a span, whose role could be identified more efficiently and accurately. However, there is no interaction between its two subtasks except for sharing span representations, which may be one

Table 2 The experimental results on the annotation datasets of [11]. AF, OF and RF represent the F1-scores (%) of aspect extraction, opinion extraction and relation detection, respectively.

	Res14			Lap14			Res15			Res16		
Models	AF	OF	RF	AF	OF	RF	AF	OF	RF	AF	OF	RF
CMLA+C-GCN[a]	81.22	80.48	63.17	–	–	53.03	76.03	74.67	55.76	–	–	62.70
RINANTE+IOG[a]	81.34	83.33	67.74	–	–	57.10	73.38	75.40	59.16	–	–	–
SpanMlt[b]	84.26	84.11	72.72	80.78	79.71	65.75	77.71	78.47	61.06	80.95	84.92	69.58
GTS[a]	83.82	85.04	75.53*	–	–	65.67	78.22	79.31	67.53	–	–	74.62
GTS	83.10	84.49	74.65	78.73	77.86	64.61	78.11	78.25	68.29	82.31	84.30	74.31
SDRN	84.70	84.01	73.08	78.28	76.48	63.63	76.30	79.71	68.00	78.93	**86.22**	75.02
STER	**85.85**	**85.89**	**74.96**	**81.06**	**81.03**	**67.64**	**80.44**	**80.91**	**69.30**	**83.64**	85.66	**75.89**

[a]The results are retrieved from [20]

[b]The results are retrieved from [19]. SpanMlt does not publish its source code, so we only compare the results copied from their released paper

* The result is better than STER, but its corresponding reproduced result is worse than STER
Results in bold are the best results in the comparison

of the main reasons why its relation detection performance is limited. Even with small entity granularity, STER still outperforms SpanMlt in the entity recognition task. Moreover, the relation extraction performance of STER is also higher than all baseline models. These results all benefit from the valuable and targeted information as well as the sufficient interaction between subtasks.

To prove that STER adapts to different annotation strategies, we also conduct experiments on the annotation datasets of [18], and the results are shown in Table 3. It is observed that STER outperforms SpanMlt and SDRN on both annotation versions, which indicates that STER is also applicable to the datasets containing sentences without aspect-opinion pairs. The proposers of SpanMlt have pointed out in their paper that the performance of SpanMlt is limited in such a dataset that contains some sentences without pair-wise relations. Removing the particular limitation from SpanMlt, we can further prove the importance of interaction for AOPE by comparing with RNCRF, CMLA, and RINANTE+SDRN-RD, which lack the interaction between entity recognition and relation detection.

In addition, although our network looks complex, the training speed is not slow. STER is trained on the GeForce GTX 1080 Ti. During the training phase, it executes 20 batches taking about 4 seconds, while SDRN takes about 3 seconds. As we discussed in Section 3.4.3, after switching the input of each time step from sentence level to token level, the complexity of the network is still within acceptable limits when the number of tracking chains is set to 100.

4.6 Ablation study

To investigate the effect of relation gate, parallel architecture, and syntactic information on STER, we conduct ablation studies on AOPE task. Table 4 reports the experimental results on the annotation datasets of [11]. While removing the relation gate, we adopt a similar algorithm to calculate the correlated degree after passing the token sequence through the changed network. This ablation model obviously runs in a pipeline method. When the parallel structure is removed, we use the hidden state at each time step as the final representation of its corresponding token. Furthermore, because our relation gate can not be set up without the parallel architecture, we conduct ablation studies in a superimposed manner. For example, in the '- syntax' experiment, the relation gate and the parallel architecture are also removed, and only the standard LSTM is retained. After removing the three critical components in turn, it is observed that the performance of our model degrades at each stage. It is reasonable because STER can achieve interaction of subtasks and learn targeted information with the parallel architecture. Moreover, the information derived from the relation gate and syntax can provide more clues for AOPE.

4.7 Case study

To clearly verify the importance of interaction between subtasks and the effect of STER, some prediction results of SDRN, STER, and the ablation model removing three key components are presented in Table 5. In the first three reviews, the ablation model fails to identify gold relations or entities due to the lack of interaction. For example, the results of relation detection can provide clues for SDRN and STER to identify the role of *'away'* in the third review, while the ablation model can not exploit the clues. Furthermore, with the help of entity tracking and rich information, STER can capture more distant relations than other models, such

Table 3 The experimental results on the annotation datasets of [18]

	Res14			Lap14			Res15		
Models	AF	OF	RF	AF	OF	RF	AF	OF	RF
RNCRF[c]	84.93	84.11	–	78.42	79.44	–	67.74	67.62	–
CMLA[c]	85.29	83.18	–	77.80	80.17	–	70.73	73.68	–
SpanMlt[c]	85.24	85.79	–	77.87	80.51	–	71.07	75.02	–
RINANTE+SDRN-RD[d]	86.45	85.67	74.34	80.16	81.96	64.17	69.90	72.09	65.42
SDRN[d]	89.49*	87.84	76.48	83.67	82.25	67.13	74.05	79.65	70.94
SDRN	88.89	87.85	76.83	82.86	81.90	68.66	75.00	79.72	70.74
STER	**89.31**	**88.27**	**78.45**	**84.06**	**82.32**	**70.80**	**75.79**	**80.55**	**72.12**

[c]The results are copied from their released paper

[d]The results are retrieved from [18]

* The result is better than STER, but its corresponding reproduced result is worse than STER
Results in bold are the best results in the comparison

Table 4 Ablation study on the annotation datasets of [11]

	Res14			Lap14			Res15		
Models	AF	OF	RF	AF	OF	RF	AF	OF	RF
STER	**85.85**	**85.89**	**74.96**	**81.06**	**81.03**	**67.64**	**80.44**	80.91	**69.30**
- relation	84.34	85.43	74.39	80.80	79.18	65.70	79.63	**80.93**	68.59
- parallel	85.48	84.99	73.73	80.68	77.44	66.48	78.63	79.29	68.44
- syntax	84.35	84.13	73.29	80.79	77.38	65.23	78.59	78.36	67.32

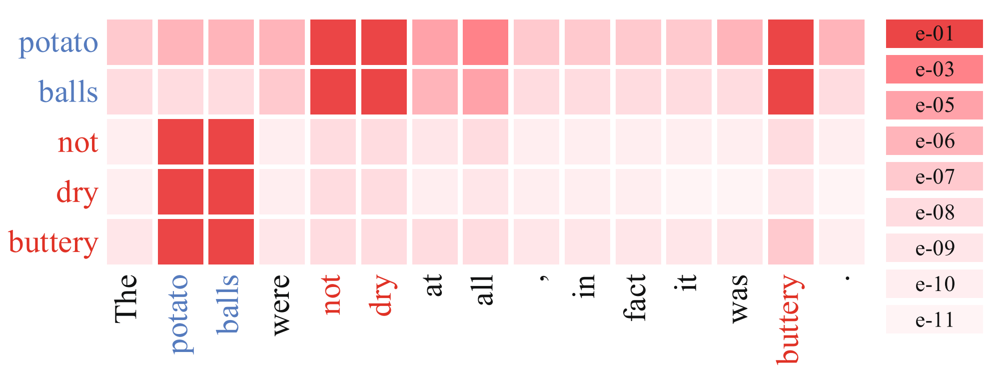
Results in bold are the best results in the comparison

Table 5 Some predictions for AOPE in the test set of Res15 which annotated by [11]

Reviews	SDRN	Ablation Model	STER
The potato balls were not dry at all, in fact it was buttery.	(potato balls, not dry) ✓ (potato balls, buttery) ✓	(potato balls, not dry) ✓ (-, buttery) ×	(potato balls, not dry) ✓ (potato balls, buttery) ✓
I attended a holiday dinner at the restaurant, and the food was majorly disappointing.	(food, disappointing) ✓	(restaurant, disappointing) × (food, disappointing) ✓	(food, disappointing) ✓
The place is a bit hidden away, but once you get there, it's all worth it.	(place, hidden away) ✓ (-, worth) ×	(place, hidden) × (-, worth) ×	(place, hidden away) ✓ (place, worth) ✓
Great place, great value.	(place, great) ✓ (value, great) ×	(place, great) ✓ (value, great) ×	(place, great) ✓

The golden aspect and opinion terms in the reviews are colored as blue and red, respectively. The correct predictions are ticked, while the wrong predictions are crossed

Fig. 4 Visualization of correlated degree, where different color intensities represent different orders of magnitude. The golden aspect and opinion terms in the reviews are colored as blue and red, respectively



as the relation between *'place'* and *'worth'* in the third review. Additionally, the extraction results of the last review indicate that STER adapts to the annotation strategy well. (*'great'*, *'value'*) is an aspect-opinion pair in the general idea, but not in the strategy of [11].

To further understand what STER has learned, we visualize the correlated degree between entities in Fig. 4. It is observed that STER accurately detects all aspect-opinion relations with the highest correlated degree and can even capture the distant relation between *'potato balls'* as well as *'buttery'*.

5 Conclusion

In this paper, we propose a model for **S**ynchronously **T**racking **E**ntities and **R**elations (STER) to deal with Aspect-Opinion Pair Extraction (AOPE) task. Inspired by Recurrent Entity Networks (EntNet), we adopt a parallel architecture to track all entities in a review. And then, we integrate a designed relation gate with Ordered Neurons LSTM (ON-LSTM) to utilize contextual, syntactic and relational information to assist in both subtasks of AOPE, i.e., entity recognition and relation detection. Moreover, the two subtasks can achieve sufficient interaction in STER due to their high dependence during calculation. The experimental results show that STER outperforms the advanced models of AOPE and adapts to different annotation strategies of the same datasets. Of course, STER still has room for further study. Following the structure of ON-LSTM and avoiding an overly complex model, we designed a unidirectional network. In the future, we will continue to study whether a bidirectional network can help improve the performance of STER when the complexity of our network is reduced.

Acknowledgements This work is supported by the National Natural Science Foundation of China under grant No.61872163 and 61806084, Jilin Province Key Scientific and Technological Research and Development Project under grant No.20210201131GX, and Jilin Provincial Education Department project under grant No.JJKH20190160KJ.

Author Contributions Yue Zhang: Conceptualization, Methodology, Software, Visualization, Writing, Editing. Tao Peng: Methodology, Supervision, Funding acquisition, Reviewing, Validation. Ridong Han: Editing, Software, Visualization. Lin Yue: Editing, Reviewing. Jiayu Han: Editing, Reviewing. Lu Liu: Reviewing, Supervision, Validation, Funding acquisition.

Funding This work is supported by the National Natural Science Foundation of China under grant No.61872163 and 61806084, Jilin Province Key Scientific and Technological Research and Development Project under grant No.20210201131GX, and Jilin Provincial Education Department project under grant No.JJKH20190160KJ.

Availability of Data and Materials All data and materials generated or analyzed during this study are included in this article.

Code Availability The authors can provide their code to editors and reviewers during the review and publish the source code when this article becomes public.

Declarations

Ethics Approval Not applicable.

Consent to Participate Not applicable.

Consent for Publication Not applicable.

Conflict of Interests The authors have no conflicts of interest to declare that are relevant to the content of this article.

References

1. Pang B, Lee L (2007) Opinion mining and sentiment analysis. Found Trends Inf Retr 2(1-2):1–135. https://doi.org/10.1561/1500000011
2. Liu B, Zhang L (2012) A survey of opinion mining and sentiment analysis. In: Mining Text Data. Springer, pp 415–463. https://doi.org/10.1007/978-1-4614-3223-4_13
3. Liu B (2012) Sentiment analysis and opinion mining, Synthesis Lectures on Human Language Technologies. Morgan & Claypool Publishers. https://doi.org/10.2200/S00416ED1V01Y201204HLT016
4. Kumar AVM, Nandkumar AN (2020) A survey on challenges and research opportunities in opinion mining. SN Comput Sci 1(3):171. https://doi.org/10.1007/s42979-020-00149-4
5. Rana TA, Cheah Y-N (2016) Aspect extraction in sentiment analysis: comparative analysis and survey. Artif Intell Rev 46(4):459–483. https://doi.org/10.1007/s10462-016-9472-z
6. Do HH, Prasad PWC, Maag A, Alsadoon A (2019) Deep learning for aspect-based sentiment analysis: A comparative review. Expert Syst Appl 118:272–299. https://doi.org/10.1016/j.eswa.2018.10.003
7. Rana TA, Cheah Y-N (2017) A two-fold rule-based model for aspect extraction. Expert Syst Appl 89:273–285. https://doi.org/10.1016/j.eswa.2017.07.047
8. Marcacini RM, Rossi RG, Matsuno IP, Rezende SO (2018) Cross-domain aspect extraction for sentiment analysis: A transductive learning approach. Decis Support Syst 114:70–80. https://doi.org/10.1016/j.dss.2018.08.009
9. Luo H, Li T, Liu B, Wang B, Unger H (2019) Improving aspect term extraction with bidirectional dependency tree representation. IEEE ACM Trans Audio Speech Lang Process 27(7):1201–1212. https://doi.org/10.1109/TASLP.2019.2913094
10. Chauhan GS, Meena YK, Gopalani D, Nahta R (2020) A two-step hybrid unsupervised model with attention mechanism for aspect extraction. Expert Syst Appl 161:113673. https://doi.org/10.1016/j.eswa.2020.113673
11. Fan Z, Wu Z, Dai X-Y, Huang S, Chen J (2019) Target-oriented opinion words extraction with target-fused neural sequence labeling. In: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Volume 1 (Long and Short Papers), Minneapolis, pp 2509–2518. https://doi.org/10.18653/v1/n19-1259
12. Veyseh APB, Nouri N, Dernoncourt F, Dou D, Nguyen TH (2020) Introducing syntactic structures into target opinion word extraction with deep learning. In: Proceedings of

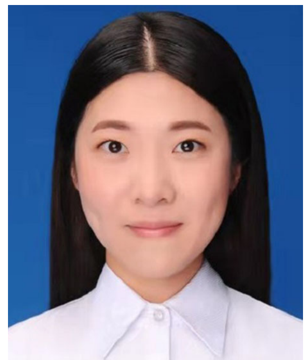
the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, pp 8947–8956. https://doi.org/10.18653/v1/2020.emnlp-main.719
13. Zhang J, Li F, Zhang Z, Xu G, Wang Y, Wang X, Zhang Y (2021) Integrate syntax information for target-oriented opinion words extraction with target-specific graph convolutional network. Neurocomputing 440:321–335. https://doi.org/10.1016/j.neucom.2020.07.152
14. Katiyar A, Cardie C (2016) Investigating lstms for joint extraction of opinion entities and relations. In: Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, ACL 2016, Volume 1: Long Papers, Berlin, pp 919–929. https://doi.org/10.18653/v1/p16-1087
15. Li X, Lam W (2017) Deep multi-task learning for aspect term extraction with memory interaction. In: Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, EMNLP 2017, Copenhagen, pp 2886–2892. https://doi.org/10.18653/v1/d17-1310
16. Yu J, Jiang J, Xia R (2019) Global inference for aspect and opinion terms co-extraction based on multi-task neural networks. IEEE ACM Trans Audio Speech Lang Process 27(1):168–177. https://doi.org/10.1109/TASLP.2018.2875170
17. Dai H, Song Y (2019) Neural aspect and opinion term extraction with mined rules as weak supervision. In: Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Volume 1: Long Papers, Florence, pp 5268–5277. https://doi.org/10.18653/v1/p19-1520
18. Chen S, Liu J, Wang Y, Zhang W, Chi Z (2020) Synchronous double-channel recurrent network for aspect-opinion pair extraction. In: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, pp 6515–6524. https://doi.org/10.18653/v1/2020.acl-main.582
19. Zhao H, Huang L, Zhang R, Lu Q, Xue H (2020) Spanmlt: A span-based multi-task learning framework for pair-wise aspect and opinion terms extraction. In: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, pp 3239–3248. https://doi.org/10.18653/v1/2020.acl-main.296
20. Wu Z, Ying C, Zhao F, Fan Z, Dai X, Xia R (2020) Grid tagging scheme for aspect-oriented fine-grained opinion extraction. In: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: Findings, pp 2576–2585. arXiv:2010.04640
21. Wu S, Fei H, Ren Y, Ji D, Li J (2021) Learn from syntax: Improving pair-wise aspect and opinion terms extraction with rich syntactic knowledge. In: Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI 2021, Virtual Event / Montreal, pp 3957–3963. https://doi.org/10.24963/ijcai.2021/545
22. Wu S, Fei H, Ren Y, Li B, Li F, Ji D (2021) High-order pair-wise aspect and opinion terms extraction with edge-enhanced syntactic graph convolution. IEEE ACM Trans Audio Speech Lang Process 29:2396–2406. https://doi.org/10.1109/TASLP.2021.3095672
23. Devlin J, Chang M-W, Lee K, Toutanova K (2019) BERT: pre-training of deep bidirectional transformers for language understanding. In: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Volume 1 (Long and Short Papers), Minneapolis, pp 4171–4186. https://doi.org/10.18653/v1/n19-1423
24. Lafferty JD, McCallum A, Pereira FCN (2001) Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In: Proceedings of the Eighteenth International Conference on Machine Learning (ICML 2001). Williams College, Williamstown, pp 282–289. http://ssli.ee.washington.edu/courses/ee517sp07/papers/crf.pdf
25. Hu M, Liu B (2004) Mining and summarizing customer reviews. In: Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Seattle, pp 168–177. https://doi.org/10.1145/1014052.1014073
26. Popescu A-M, Etzioni O (2005) Extracting product features and opinions from reviews. In: HLT/EMNLP 2005, human language technology conference and conference on empirical methods in natural language processing, proceedings of the conference, Vancouver, pp 339–346. https://www.aclweb.org/anthology/H05-1043/
27. Zhuang L, Jing F, Zhu X (2006) Movie review mining and summarization. In: Proceedings of the 2006 ACM CIKM International Conference on Information and Knowledge Management, Arlington, pp 43–50. https://doi.org/10.1145/1183614.1183625
28. Qiu G, Liu B, Bu J, Chen C (2011) Opinion word expansion and target extraction through double propagation. Comput Linguist 37(1):9–27. https://doi.org/10.1162/coli_a_00034
29. Jakob N, Gurevych I (2010) Extracting opinion targets in a single and cross-domain setting with conditional random fields. In: Proceedings of the 2010 conference on empirical methods in natural language processing, EMNLP 2010, A meeting of sigdat, a special interest group of the ACL. MIT Stata Center, Massachusetts, pp 1035–1045. https://www.aclweb.org/anthology/D10-1101/
30. Li F, Han C, Huang M, Zhu X, Xia Y, Zhang S, Yu H (2010) Structure-aware review mining and summarization. In: COLING 2010, 23rd international conference on computational linguistics, proceedings of the conference, Beijing, pp 653–661. https://www.aclweb.org/anthology/C10-1074/
31. Poria S, Cambria E, Gelbukh AF (2016) Aspect extraction for opinion mining with a deep convolutional neural network. Knowl Based Syst 108:42–49. https://doi.org/10.1016/j.knosys.2016.06.009
32. Wang W, Pan SJ, Dahlmeier D, Xiao X (2016) Recursive neural conditional random fields for aspect-based sentiment analysis. In: Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, EMNLP 2016, Austin, pp 616–626. https://doi.org/10.18653/v1/d16-1059
33. Wang W, Pan SJ, Dahlmeier D, Xiao X (2017) Coupled multi-layer attentions for co-extraction of aspect and opinion terms. In: Proceedings of the thirty-first AAAI conference on artificial intelligence, Francisco, pp 3316–3322. http://aaai.org/ocs/index.php/AAAI/AAAI17/paper/view/14441
34. Chen Z, Qian T (2020) Relation-aware collaborative learning for unified aspect-based sentiment analysis. In: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, pp 3685–3694. https://doi.org/10.18653/v1/2020.acl-main.340
35. Gao L, Wang Y, Liu T, Wang J, Zhang L, Liao J (2021) Question-driven span labeling model for aspect-opinion pair extraction. In: Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021, Thirty-Third Conference on Innovative Applications of Artificial Intelligence, IAAI 2021, The Eleventh Symposium on Educational Advances in Artificial Intelligence, EAAI 2021, Virtual Event, pp 12875–12883. https://ojs.aaai.org/index.php/AAAI/article/view/17523
36. Henaff M, Weston J, Szlam A, Bordes A, LeCun Y (2017) Tracking the world state with recurrent entity networks. In: 5th International Conference on Learning Representations, ICLR 2017, Conference Track Proceedings, Toulon. https://openreview.net/forum?id=rJTKKKqeg
37. Liu F, Cohn T, Baldwin T (2018) Recurrent entity networks with delayed memory update for targeted aspect-based sentiment analysis. In: Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics:

Human Language Technologies, NAACL-HLT, Volume 2 (Short Papers), New Orleans, pp 278–283. https://doi.org/10.18653/v1/n18-2045
38. Shen Y, Tan S, Sordoni A, Courville AC (2019) Ordered neurons: Integrating tree structures into recurrent neural networks. In: 7th International Conference on Learning Representations, ICLR 2019, New Orleans. https://openreview.net/forum?id=B1l6qiR5F7
39. Pontiki M, Galanis D, Pavlopoulos J, Papageorgiou H, Androutsopoulos I, Manandhar S (2014) Semeval-2014 task 4: Aspect based sentiment analysis. In: Proceedings of the 8th International Workshop on Semantic Evaluation, SemEval@COLING 2014, Dublin, pp 27–35. https://doi.org/10.3115/v1/s14-2004
40. Pontiki M, Galanis D, Papageorgiou H, Manandhar S, Androutsopoulos I (2015) Semeval-2015 task 12: Aspect based sentiment analysis. In: Proceedings of the 9th International Workshop on Semantic Evaluation, SemEval@NAACL-HLT 2015, Denver, pp 486–495. https://doi.org/10.18653/v1/s15-2082
41. Pontiki M, Galanis D, Papageorgiou H, Androutsopoulos I, Manandhar S, Al-Smadi M, Al-Ayyoub M, Zhao Y, Qin B, Clercq OD, Hoste V, Apidianaki M, Tannier X, Loukachevitch NV, Kotelnikov EV, Bel N, Zafra SMJ, Eryigit G (2016) Semeval-2016 task 5: Aspect based sentiment analysis. In: Proceedings of the 10th International Workshop on Semantic Evaluation, SemEval@NAACL-HLT 2016, San Diego, pp 19–30. https://doi.org/10.18653/v1/s16-1002
42. Zhang Y, Qi P, Manning CD (2018) Graph convolution over pruned dependency trees improves relation extraction. In: Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, pp 2205–2215. https://doi.org/10.18653/v1/d18-1244

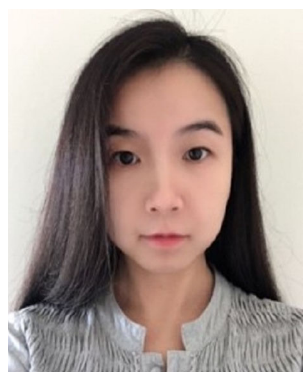
Yue Zhang received the BE degree in Computer Science and Technology from Yanbian University, China, in 2019. She is currently working toward the ME degree in the College of Computer Science and Technology at Jilin University, China. Her research interests include sentiment analysis, information extraction and applications of deep learning to NLP.

Tao Peng received the PhD degree in Computer Science from Jilin University, China, in 2007. He is a professor in the College of Computer Science and Technology, and the vice-director of the Institute of Big Data Science and Technology at Jilin University, China. His research interests include machine learning, data mining, natural language processing and applications of deep learning to NLP.

Ridong Han received the BS degree in Computer Science and Technology from Jilin University, China, in 2018. He is currently working toward the PhD degree in the College of Computer Science and Technology at Jilin University, China. His research interests include knowledge graph, information extraction and applications of deep learning to NLP.



Jiayu Han received the BE degree in Computer Science and Technology, ME degree in Computer Software and Theory, and PhD degree in Computer Application Technology from Jilin University, China, in 2012, 2015, and 2020. She is currently pursuing an MS degree in Computational Linguistics at the University of Washington, USA. Her research interests include natural language processing, knowledge graph, and recommender systems.



Lin Yue received the BS degree in Computer Science and Technology, MS degree in Computer Application Technology from Northeast Normal University, China, and PhD degree in Computer Application Technology from Jilin University, China, which is jointly supervised by Prof Xue Li (University of Queensland) and Prof Wangli Zuo (Jilin University). She is currently working in the School of Information Technology and Electrical Engineering at the University of Queensland, Australia. Her research interests include data mining, machine learning, health data analytics, and brain-computer interface.

Lu Liu received the BE degree in Computer Science and Technology and PhD degree in Computer Science from Jilin University, China, in 2012 and 2017. She is currently a lecturer in the College of Software at Jilin University, China. Her research interests include machine learning, data mining, natural language processing and applications of deep learning to NLP.

Affiliations

Yue Zhang[1,3] · Tao Peng[1,2,3] · Ridong Han[1,3] · Jiayu Han[4] · Lin Yue[5] · Lu Liu[1,2,3]

Yue Zhang
yue_zhang19@mails.jlu.edu.cn

Ridong Han
hanrd20@mails.jlu.edu.cn

Jiayu Han
jyhan126@uw.edu

Lin Yue
l.yue@uq.edu.au

Lu Liu
liulu@jlu.edu.cn

1 College of Computer Science and Technology, Jilin University, Changchun, Jilin 130012, China

2 College of Software, Jilin University, Changchun, Jilin 130012, China

3 Key Laboratory of Symbol Computation and Knowledge Engineer of the Ministry of Education, Changchun, Jilin 130012, China

4 Department of Linguistics, University of Washington, Seattle, WA 98195, USA

5 School of Information Technology and Electrical Engineering, University of Queensland, Brisbane, QLD 4072, Australia