Semantic and Syntactic Enhanced Aspect Sentiment Triplet Extraction

Zhexue Chen[1,2,3], **Hong Huang**[1,2,3*], **Bang Liu**[4*], **Xuanhua Shi**[1,2,3], **Hai Jin**[1,2,3]
[1]National Engineering Research Center for Big Data Technology and System
[2]Service Computing Technology and System Lab
[3]Huazhong University of Science and Technology, China
[4]RALI & Mila, University of Montreal
{chenzhexue,honghuang,xhshi,hjin}@hust.edu.cn
bang.liu@umontreal.ca

Abstract

Aspect Sentiment Triplet Extraction (ASTE) aims to extract triplets from sentences, where each triplet includes an entity, its associated sentiment, and the opinion span explaining the reason for the sentiment. Most existing research addresses this problem in a multi-stage pipeline manner, which neglects the mutual information between such three elements and has the problem of error propagation. In this paper, we propose a *Semantic and Syntactic Enhanced aspect Sentiment triplet Extraction model* (S^3E^2) to fully exploit the syntactic and semantic relationships between the triplet elements and jointly extract them. Specifically, we design a Graph-Sequence duel representation and modeling paradigm for the task of ASTE: we represent the semantic and syntactic relationships between word pairs in a sentence by graph and encode it by *Graph Neural Networks* (GNNs), as well as modeling the original sentence by LSTM to preserve the sequential information. Under this setting, we further apply a more efficient inference strategy for the extraction of triplets. Extensive evaluations on four benchmark datasets show that S^3E^2 significantly outperforms existing approaches, which proves our S^3E^2's superiority and flexibility in an end-to-end fashion.

1 Introduction

Aspect-based Sentiment Analysis (ABSA) usually requires to extract comment targets in a review and judge corresponding sentiment polarities (Liu, 2012; Pontiki et al., 2014). Such a research field has received widespread attention (Zhang et al., 2015; Li and Lu, 2017, 2019; Li et al., 2019a). In this paper, we concentrate on a more relatively fine-grained task - *Aspect Sentiment Triplet Extraction* (ASTE) (Peng et al., 2020), which aims to extract triplets, including aspects (e.g., entities), the corresponding sentiment for each aspect, and the opinion spans explaining the reason for the sentiment. An example is shown in Fig. 1. It contains two triplets, $(Waiters, friendly, +)$ and $(fruit\ salad, so\ so, 0)$ where we use +, -, and 0 to represent positive, negative, and neutral sentiment. Unlike the ABSA task that extracts two tuples, $(Waiters, +)$ and $(fruit\ salad, 0)$ in this sentence, such triplets extracted by ASTE task can better reflect multiple emotional factors (aspect, opinion, sentiment) from the user reviews and are more suitable for practical application scenarios.

The ASTE task is extremely challenging because it requires extracting these three elements in one shot. Straightforwardly, one naive solution is to split the ASTE task into two stages in a pipeline manner using a unified tagging schema [1] (Peng et al., 2020). Such a pipeline approach lacks an effective mechanism to capture the three elements' relationship and suffers from error propagation. Another solution for the ASTE task is to use an end-to-end model to extract triplets (Xu et al., 2020; Wu et al., 2020). Yet, these methods focus on designing a new tagging schema to formalize ASTE into a unified task and cannot effectively establish the connection between words and ignore the semantic and syntactic relationship between the three elements.

Besides, a sentence may contain a one-to-many case, that is, one aspect corresponds to multiple opinions, or one opinion corresponds to multiple aspects. For instance, in the sentence *"We love the food, drinks, and atmosphere,"* the opinion "love" is associated with three aspects "food", "drinks", and "atmosphere". This situation is quite common in reality, increasing the difficulty of match-

*Corresponding Author

[1]It consists of $\{B, I, E, S\} - \{NEU, NEG, POS\}$ and tag O, which denote beginning, inside, end, single-word target with neutral, negative and positive sentiment respectively and outside of a target.

Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021, pages 1474–1483

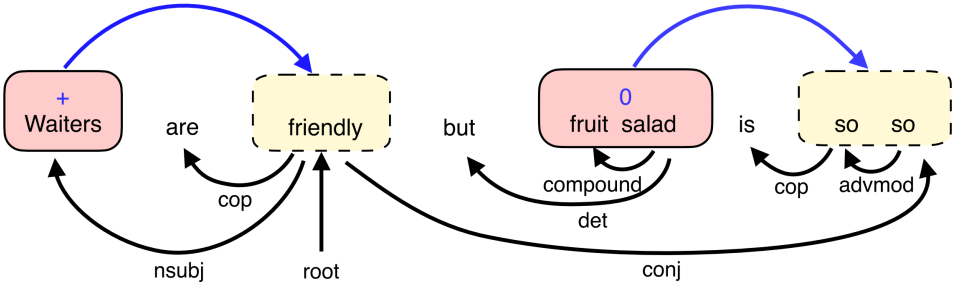

Figure 1: An example of the ASTE task. The words in the solid and dashed boxes are aspects and opinions, respectively. The blue arrows above represent the correspondence between them. The black arrows below represent the dependencies between words.

ing aspects with opinions. Nevertheless, current solutions either fail to capture these one-to-many relationships (Xu et al., 2020) or ignore the semantic relationship between word pairs in a triplet (Wu et al., 2020).

Furthermore, various relationships exist among triplets, such as syntactic dependence and semantic word similarity, which have been neglected. For example, as shown in Figure 1, there is a nominal subject dependency (called $nsubj$) between $waiters$ and $friendly$, indicating that there exists an aspect. Also, the two opinions, $friendly$ and $so\ so$ in the sentence are associated with each other, where there is a conjunct dependency (called $conj$), implying they have similar attributes.

To fully utilize these implicit relationships, we design a ***S**emantic and **S**yntactic **E**nhanced Aspect **S**entiment Triplet **E**xtraction model* (S^3E^2). S^3E^2 utilizes semantic and syntactic information from words, which helps to distinguish words' attributes and identify the relationship between word pairs. In order to better leverage these relationships, we build a *Graph Neural Network* (GNN) based model to capture the interactions between words and triplet elements. For each sentence, we transform it into a unique text graph representation, where each node is a word, and the edges are established based on attention to the words themselves, adjacent relationships, and syntactic dependencies. Such a concise and effective text graph can obtain the precise meaning of each word and gain insight into their relations.

Moreover, we further utilize LSTM (Hochreiter and Schmidhuber, 1997) to learn the contextual semantics of each word from a sequential perspective, forming a Graph-Sequence duel modeling of a sentence. In this way, S^3E^2 has an excellent ability to distinguish the categories of words and more accurately recognize the relationship between word pairs. With the semantic and syntactic enhanced module, the correlation between word pairs is well captured, yielding a more simple inference strategy for triplet extraction. Since S^3E^2 can perceive the semantics and syntax from words excellently, we only need to infer once for all datasets to obtain more accurate triplets and save time overhead. Finally, we parse out the triplets from the final predictions.

We run extensive experiments on four benchmark datasets. The experimental results show that S^3E^2 achieves significantly better performance than existing state-of-the-art approaches by fully exploiting the syntactic and semantic 18 relationships between word pairs.

To summarize, our main contributions include the following:

- We design a graph representation of a sentence which integrates the syntactic dependency, semantic relatedness, and positional relationship between words, and encode it with Graph Neural Networks to fully exploit the various correlations.
- We further model the sentence with LSTM to incorporate its sequential information, forming a Graph-Sequence duel modeling paradigm. Moreover, we only need to infer once for all datasets, demonstrating the superiority of S^3E^2.
- We make extensive experiments, and the results show S^3E^2 outperforms all state-of-the-art approaches significantly for triplet extraction.

2 Our Approach

We design an effective framework to complete triplet extraction in an end-to-end fashion. The overall model architecture is shown in Figure 2. In this section, we first define the ASTE task, describe the Grid Tagging Schema and deconstruct triplets

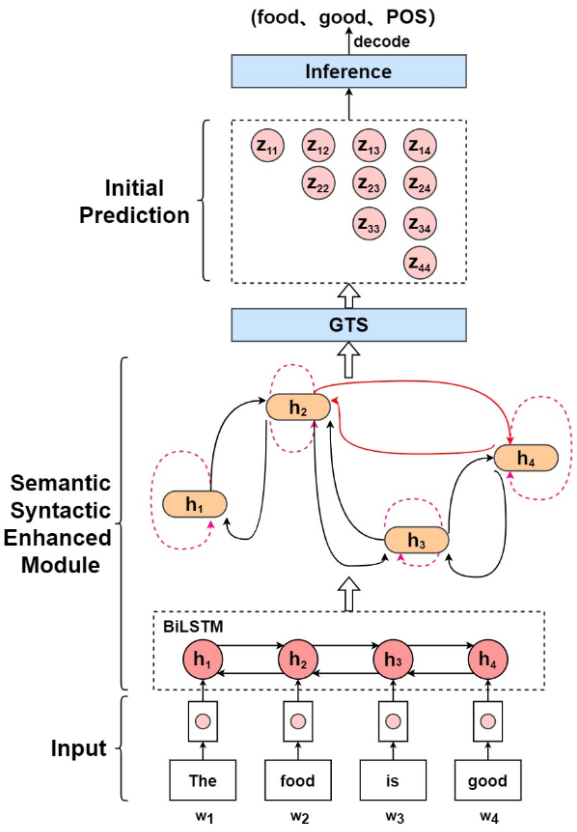

Figure 2: The overall architecture of our end-to-end model S^3E^2. In our text graph, the type of dashed edges is self-loop, the type of black solid edges is neighbor edge, and the type of red solid edge is dependent edge.

from it in detail. We next present S^3E^2 model, followed by our inference strategy.

2.1 Task Definition and Preliminaries

Definition: Triplet Extraction. Given an input sentence $x = \{x_1, x_2, \cdots, x_n\}$ with length n, each word has two tag labels: the aspect tag label and the opinion tag label, respectively. Their tagging schema is $\mathcal{Y}$ = {B, I, O}, denoting the beginning, inside, outside of one aspect term or opinion term. Meanwhile, each aspect target is annotated with a sentiment polarity label $\mathcal{S}$ = {NEU, POS, NEG}, denoting neutral, positive, and negative sentiment expressed towards itself. Our goal is to extract a set of triplets $\mathcal{T} = \{(a, o, s)_m\}_{m=1}^{|\mathcal{T}|}$ from the sentence x, where the notations a, o, and s stand for an aspect, an opinion, and corresponding sentiment polarity, respectively. The notation $(a, o, s)_m$ is a triplet in x and $|\mathcal{T}|$ represents the total number of triplets in this sentence.

Grid Tagging Schema. To tackle the ASTE task, a *Grid Tagging Schema* (GTS) was proposed

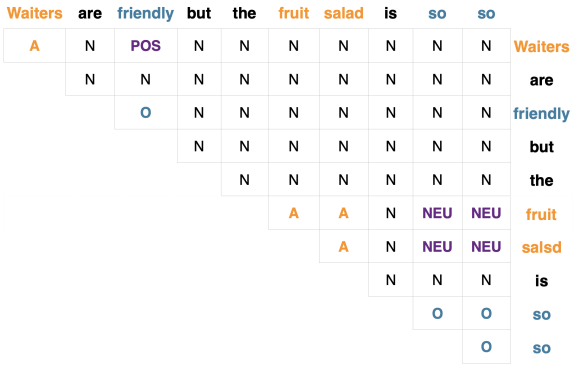

Figure 3: A tagging example for GTS

by Wu et al. (2020), which adopts six tags $\mathcal{G} = \{\text{A}, \text{O}, \text{NEG}, \text{NEU}, \text{POS}, \text{N}\}$ to represent the relationship for any pair of two words (w_i, w_j) in a sentence. The two tags, A and O, denote the word-pair (w_i, w_j) is the same aspect or opinion, respectively. The three tags NEG, NEU, POS denote negative, neutral, or positive emotions expressed for the triplet consisting of the pair of words (w_i, w_j) that exactly contains an aspect term and an opinion term. The tag N denotes non above relations for word-pair (w_i, w_j). A tagging example is shown in Figure 3. In detail, the three coordinates in the grid $(5, 5)$, $(6, 6)$, and $(6, 5)$ respectively form word pairs $(fruit, fruit)$, $(salad, salad)$, and $(fruit, salad)$, which are labeled A because they all belong to the same aspect. The same logic applies to opinions. The coordinate $(2, 0)$ is labeled POS because it makes a correct triplet $(Waiters, friendly, POS)$, which contains exactly the right aspect, opinion, and sentiment information. For simplicity, we use an upper triangular grid.

Triplets Decoding. we explain how to decode triplets based on the predicted grid tags. We take the decoding algorithm designed by Wu et al. (2020). First, both aspects and opinions were identified using the predictive tags of all word pairs (w_i, w_j) on the main diagonal without considering other word pairs' constraints. The span consisting of continuous A is regarded as a complete aspect, and the span consisting of continuous O is detected as a complete opinion. At this point, we have extracted the aspect a and opinion o. Then, we count the predicted tags of all word pairs (w_i, w_j) when $w_i \in a$ and $w_j \in o$. The most predictive sentiment label $s \in S$ is regarded as sentiment polarity for triplet (a, o, s). When there are multiple most predictive sentiment labels, then the label is decided

by the order: positive > neutral > negative. If they are all predicted to be label N, we consider that a and o cannot constitute a triplet.

2.2 Semantic and Syntactic Enhanced ASTE Model

Since this task requires extracting multiple elements from a sentence, it is important to design a model that can effectively distinguish the properties of words and master the relationship between them. S^3E^2 first uses LSTM to encode sentences so that we can perceive contextual semantic. In order to capture many-sided features, S^3E^2 next applies graph neural network to model syntactic dependency, semantic relatedness, and positional relationship between words. Finally, an inference strategy is proposed, which only makes one inference to further extract more accurate triplets for all datasets.

2.2.1 Graph-Sequence Duel Representation

We first apply a bidirectional *Long Short Term Memory* (LSTM) networks (Hochreiter and Schmidhuber, 1997) to encode the input sentence x. LSTM is capable of learning contextual semantic representation since it can mark key semantics from previous time steps. Hence, we learn contextual features $\{\mathbf{h}_1, \mathbf{h}_2, \cdots, \mathbf{h}_n\}$ for the input sequence.

We observe that different words in a sentence often have various internal relationships. As elaborated in Figure 1, there is a syntactic dependency between $waiters$ and $friendly$, since opinions often modify aspects. Besides, words that are semantically similar may also be related. The two opinions, $friendly$ and $so\ so$, although they are far apart, there is still a dependency between them. Therefore, it is of great help to model the relationships and grasp semantic and syntactic information from words. With this in mind, we build a unique text graph for every input sentence using graph neural network.

Formally, a text graph $G = (V, E)$ is a structure used to represent words and their relations, which consists of the set of nodes V and the set of edges E. Each word in the sentence is regarded as a node, while the relationships between words are considered edges. We construct three types of edges: self-loop edge, neighbor edge, and dependency edge. If there is an edge connecting to the node itself, then the edge is the self-loop edge. The edge connecting a node and its neighbor is a neighbor edge, while if there exists a dependency relationship between two nodes, then there is a dependency edge between them. Specifically, we define the text graph as follows:

$$V = \{v_i \mid i \in [1, n]\} \tag{1}$$

$$E = \{e_{ij} \mid j = [i-1, i+1] \cup D_i\} \tag{2}$$

where D_i represents a set of nodes with which node v_i has a dependency. All edges are bidirectional and the node feature for v_i is taken from $\mathbf{h}_i$. We adopt GraphSAGE (Hamilton et al., 2017) to generate representations $\left\{\widetilde{\mathbf{h}}_1, \widetilde{\mathbf{h}}_2, \cdots, \widetilde{\mathbf{h}}_n\right\}$ for each node. We chose LSTM aggregator from GraphSAGE because it has stronger expressive ability.

Then, we concatenate the integrated representations of w_i and w_j to represent all word pairs (w_i, w_j), i.e., $\mathbf{r}_{ij} = \left[\widetilde{\mathbf{h}}_i; \widetilde{\mathbf{h}}_j\right]$, where $[\cdot;]$ is a concatenation operation. All representations of word pairs correspond to cells in our grid, which is then fed to a linear layer to calculate initiatory probability distribution $\mathbf{z}_{ij} \in R^{|\mathcal{G}|}$ through:

$$\mathbf{z}_{ij} = \mathbf{W}_s \mathbf{r}_{ij} + \mathbf{b}_s \tag{3}$$

where $\mathbf{W}_s$ and $\mathbf{b}_s$ are trainable parameters.

2.2.2 Inference Strategy

The initial probability distribution $\mathbf{z}_{ij}$ between all word pairs obtained above can further facilitate more accurate extraction of triplets. For instance, if $(0, 0)$ and $(2, 2)$ in grid tagging example are predicted to be A and O, respectively, then the position at which they intersect $(0, 2)$ is even less likely to be predicted to be N, and vice versa. Also, since many aspects or opinions are made up of multiple words, if a certain coordinate is predicted as one of $\mathcal{S}$, then its adjacent locations are more likely to be predicted to be the same sentiment label.

Therefore, we employ an inference strategy to obtain more accurate triplets by observing the characteristics of the initial probability distributions through the below processes. Formally, new feature representation $\mathbf{g}_{ij}$ learning is as follows:

$$\begin{aligned} \mathbf{z}_i &= \text{maxpooling}\left(\mathbf{z}_{i,:}\right) \\ \mathbf{z}_j &= \text{maxpooling}\left(\mathbf{z}_{j,:}\right) \\ \widetilde{\mathbf{r}}_{ij} &= [\mathbf{r}_{ij}; \mathbf{z}_i; \mathbf{z}_j; \mathbf{z}_{ij}] \\ \mathbf{g}_{ij} &= \mathbf{W}_g \widetilde{\mathbf{r}}_{ij} + \mathbf{b}_g \end{aligned} \tag{4}$$

where $\mathbf{W}_g$ and $\mathbf{b}_g$ are trainable parameters. The symbol $[\cdot;]$ represents a concatenation operation. Concretely, $\mathbf{z}_{i,:} = (\mathbf{z}_{1:i,i}, \mathbf{z}_{i,i:n})$ because of the upper triangular grid in GTS. $\mathbf{z}_i$/$\mathbf{z}_j$ works by capturing the associated features between w_i/w_j and other words.

It is worth noting that inference strategy by Wu et al. (2020) are unable to well capture the relationship between words, thus yielding indefinite number of iterations for inference, which increases the time complexity when the number of inferences is large. In contrast, we only need to infer once for all datasets with semantic and syntactic enhanced module, which further proves the superiority of S^3E^2.

Finally, we send $\mathbf{g}_{ij}$ to a linear layer with softmax activation function for classification.

$$p_{ij} = \text{softmax}\left(\mathbf{W}_p \mathbf{g}_{ij} + \mathbf{b}_p\right) \tag{5}$$

where $\mathbf{W}_p$ and $\mathbf{b}_p$ are trainable parameters.

2.3 Training Loss Function

The training goal for the ASTE task is to minimize the cross-entropy error for all word pairs. The unified loss function is defined as:

$$\mathcal{L} = -\sum_{i=1}^{n}\sum_{i=i}^{n}\sum_{k\in\mathcal{G}} I\left(y_{ij} = k\right)\log\left(p_{i,j}\right) \tag{6}$$

where y_{ij} denotes the one-hot vector of ground truth for the word pair (w_i, w_j) and $I(\cdot)$ indicates the k-th component being 1.

3 Experiments

3.1 DataSets

We conduct experiments on four datasets integrated by Wu et al. (2020). Each dataset has been divided into three parts: training set, validation set, and test set. Table 1 lists the statistics for these datasets. 14res, 15res, and 16res belong to the restaurant domain, while 14lap is of laptop domain. Each sentence has been annotated with a sequence of aspect tags and opinion tags and sentiment polarity of corresponding aspects. These datasets originally come from SemEval Challenges (Pontiki et al., 2014, 2015, 2016).

Note that each sentence may have more than one aspect and opinion. Besides, one aspect may be associated with multiple opinions and vice versa. For 14res, 14lap, 15res, and 16res, the proportion of one-to-many data reaches 37.27%, 38.54%, 33.39%, and 33.13%, respectively. Various relationships usually exist between aspects and opinions, using them is beneficial to triplet extraction. We count the ratio of triplets with implicit relationships. For these four datasets, they are 79.37%, 74.22%, 76.27%, and 80.57%, respectively.

3.2 Baselines

We compare the performance of S^3E^2 with the following approaches, where most triplet extraction models currently are done in a pipeline manner, and few state-of-the-art models are in an end-to-end way.

- **Peng-unified-R+PD.** Peng et al. (2020) proposed a pipeline approach in two stages. The first stage model (Peng-unified-R) jointly extracts aspects with sentiment using the unified tagging schema and opinion location in the BIEOS tagging schema. It leverages mutual information between aspects and opinions. In the second stage, all candidate triplets are generated, and a MLP-based classifier (PD) is applied to determine whether each triplet is valid or not.
- **Li-unified-R+PD.** A pipeline approach combined by Peng et al. (2020). In the first stage, the model (Li et al., 2019a) is modified to co-extract aspects with sentiment as well as extracting opinion. In the second stage, it applies the same classifier (PD) mentioned above to obtain all the valid triplets.
- **Peng-unified-R+IOG.** A pipeline approach combined by Wu et al. (2020). It first employs the model Peng-unified-R of Peng et al. (2020) for extracting aspects with sentiment, then uses IOG (Fan et al., 2019) to produce final triplets. The IOG encodes the information from a given asepct to extract its opinion words.
- **IMN+IOG.** Another pipeline approach combined by Wu et al. (2020). It first employs the IMN (He et al., 2019) for extracting aspects with sentiment, then uses the IOG (Fan et al., 2019) to produce final triplets.
- **Grid.** A state-of-the-art approach model proposed by Wu et al. (2020), which designs a grid tagging schema to address triplet extraction in an end-to-end way. It employs an inference strategy to utilize the mutual indications between different opinion factors. For a fair comparison, we choose their model Grid-CNN and Grid-BiLSTM, which use CNN encoder and BiLSTM encoder respectively.

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

Dataset	14res					14lap					15res					16res				
	#S	#T	#-	#0	#+	#S	#T	#-	#0	#+	#S	#T	#-	#0	#+	#S	#T	#-	#0	#+
train	1259	2356	491	172	1693	899	1452	533	111	808	603	1038	210	29	799	863	1421	330	55	1036
val	315	580	107	46	427	225	383	136	48	199	151	239	49	9	181	216	348	77	8	263
test	493	1008	156	68	784	332	547	116	67	364	325	493	144	25	324	328	525	79	30	416

Table 2: Experimental results of triplet extraction. Best results are in bold. The mark "*" means that S^3E^2 significantly outperforms all baselines. The mark "-" means that the original code of the IMN method does not contain the resources required to run on the dataset 16res.

Model	14res			14lap			15res			16res		
	P	R	F	P	R	F	P	R	F	P	R	F
Li-unified-R+PD	41.44	68.79	51.68	42.25	42.78	42.47	43.34	50.73	46.69	38.19	53.47	44.51
Peng-unified-R+PD	44.18	62.99	51.89	40.40	47.24	43.50	40.97	54.68	46.79	46.76	62.97	53.62
Peng-unified-R+IOG	58.89	60.41	59.64	48.62	45.52	47.02	51.70	46.04	48.71	59.25	58.09	58.67
IMN+IOG	59.57	63.88	61.65	49.21	46.23	47.68	55.24	52.33	53.75	–	–	–
Grid-CNN	70.79	61.71	65.94	55.93	47.52	51.38	60.09	53.57	56.64	62.63	66.98	64.73
Grid-BiLSTM	67.28	61.91	64.49	59.42	45.13	51.30	63.26	50.71	56.29	66.07	65.05	65.56
S^3E^2	69.08	64.55	**66.74***	59.43	46.23	**52.01**	61.06	56.44	**58.66***	71.08	63.13	**66.87***

3.3 Implementation Details

Following the previous work (Wu et al., 2020), we combine a 300-dimension domain-general embedding from GloVe (Pennington et al., 2014) and pre-trained with 840 billion tokens and a 100- dimension domain-specific embedding trained with fastText (Bojanowski et al., 2017) to initialize double word embeddings for S^3E^2. The learning rate is 0.001, and the dropout rate is 0.5. We use Adam (Kingma and Ba, 2015) as S^3E^2 optimizer. The number of layer for LSTM is 1 and the cell is set to 50. The aggregator type from GraphSAGE we chose is LSTM. We use Stanza (Qi et al., 2020) to parse the dependencies in the sentence. The batch size is set to 32 for all datasets and the valid set is used for early stopping. We select the best model according to the best F1 score on the valid set and run the test set with it for evaluation.

Following previous work, we report experimental results based on precision (P), recall (R), and F1 scores. Note that the F1 score measures the performance of mating triplets, which means a triplet is correct only when the aspect span, its corresponding sentiment, and opinion span are all proper.

3.4 Main Results For Triplet Extraction

Table 2 presents the main results of the final triplet extraction. S^3E^2 surpasses all baselines significantly on all datasets. Compared with the best results of existing baselines, S^3E^2 still achieves an apparent absolute F1 scores increase of 2.02% and 1.31% on 15res and 16res, respectively, and achieved an impressive increase of 0.80% and 0.63% on 14res and 14lap, respectively. Except for Grid-CNN and Grid-BiLSTM, the other models are all pipeline methods.

The experimental results show that S^3E^2 is far beyond these methods, which also strongly proves the advantages of the semantic and syntactic enhanced model. When we compare S^3E^2 with competitive baselines, Grid-CNN and Grid-BiLSTM in detail, we find that the reason why we perform better on 14res and 15res is because we extract a more complete set of triplets in these two datasets, resulting a more significant recall. The reason why we perform better on 14lap and 16res is because we extract more accurate triplets, resulting a more significant precision. Such comprehensive results demonstrate the strength of S^3E^2, which has the ability to learn multi-faceted semantics and and is good at extracting triplets.

4 Experiment Analysis

4.1 Ablation Study

To investigate the effectiveness of different modules in S^3E^2, we conduct ablation study for the ASTE task. As shown in Table 3, S^3E^2 represents our full model that equipped with all modules. Next, we will carefully observe the role of each module by introducing four model variants, namely Dep, Infer, Graph, and BiLSTM.

Infer means removing the inference strategy

Table 3: Results of ablation study for the ASTE task

Models	14res	14lap	15res	16res
	F	F	F	F
S^3E^2	66.23	52.01	58.66	66.87
Infer	64.20	48.68	56.90	63.27
Dep	66.74	50.43	57.43	64.98
Graph	62.12	46.37	53.77	63.63
BiLSTM	62.48	44.78	54.38	61.54

Table 4: Results of triplet extraction on different aggregators and number of graph network layers

Aggregator	Layers	14res	14lap	15res	16res
		F	F	F	F
LSTM	2	64.83	47.32	55.84	62.96
	3	66.23	52.01	58.66	66.87
Mean	2	64.28	47.00	55.15	62.73
	3	64.43	50.26	54.10	63.70

from S^3E^2. We can see that F1 scores drop sharply, which shows that the inference strategy can grasp the relationship between the three elements in the triplets from the previous round of predictions to promote the ASTE task. Dep means that when constructing a text graph for a sentence, we do not add the third edge type mentioned above. We can see that F1 scores drop except for res14, showing that overall the dependent edges can help the model better master relationships. The training set of 14res is larger than other datasets. When training the full model, we may overfit due to the setting of parameters (e.g., epoch, batch size), resulting in slightly lower performance, compared with Dep.

Graph means removing the graph-based GNN modules. After removing the entire graph, the performance of the model is greatly reduced. Obviously, the graph neural networks can well perceive the relational semantics and distinguish the characteristics of the words. The F1 scores also decline sharply when we remove the BiLSTM, which shows that contextual semantic information is helpful. Comparing Graph and BiLSTM, we find that the former has higher results on 14lap and 16res. It may be that these two datasets are more dependent on contextual semantic features. In general, each module of S^3E^2 contributes to the extraction of triplets.

4.2 Effects of Aggregator Types

In order to study the impact of aggregator types on performance, we report the results of different aggregator types for the ASTE task on these four datasets in Table 4. There are two types of aggregators, LSTM and Mean, adopted from (Hamilton et al., 2017). The former is based on the LSTM structure (Hochreiter and Schmidhuber, 1997) and is applied to the random arrangement of the node's neighbors. The latter is just based on the mean operation. As shown in Table 4, when the network layers of the two aggregators are equal, no matter how many layers, the effect of the LSTM aggregator is better than that of the Mean aggregator. This phenomenon indicates that the LSTM aggregator has stronger expressive ability and is more suitable for the ASTE task.

4.3 Effects of Graph Network Layers

To examine the effects of the number of graph network layer, we also present the results of different layers on these four datasets to extract triplets. It can be observed that the experimental performance increases as the number of layers increases from 2 to 3 for the same type of aggregator. This proves that the ability of graph neural networks to gather features is related to the number of network layers. We notice that when the number of layers is set to 2, the LSTM aggregator has higher performance than the Mean aggregator by 0.55%, 0.32%, 0.69%, and 0.23% on the four datasets, respectively. Nevertheless, when the number of layers is 3, their performance differs by 1.80%, 1.75%, 4.56%, and 3.17%. As the number of layers increases, the performance gap between the LSTM aggregator and the Mean aggregator widens significantly, which further illustrates the advantage of the LSTM aggregator.

4.4 Case Study

Five typical cases are presented in Table 5. The first example is a simple case without complicated word order and all models can predict accurately. The second example comes from the restaurant field, which expresses a negative attitude tactfully. Both Grid-BiLSTM and Grid-CNN incorrectly predict sentiment for "staff", and Grid-CNN mistakenly predicts "should be" as an aspect.

The third example directly expresses negative sentiment, which is picked from the laptop field. We can observe that Grid-LSTM and Grid-CNN mistakenly regard "maintain" as an aspect, and also make a false prediction for sentiment. For these two examples, S^3E^2 makes accurate judgments, which

Table 5: Case analysis. The first column is five representative examples, the second column is golden truth, and the other columns are the output results of different models.

Example	Golden Triplets	Grid-BiLSTM	Grid-CNN	Our
The bread is top notch as well	(bread,top notch,POS)	(bread,top notch,POS)	(bread,top notch,POS)	(bread,top notch,POS)
The staff should be a bit more friendly	(staff,friendly,NEG)	(staff,more friendly,POS) (staff,should be,POS)	(staff,more friendly,POS)	(staff,friendly,NEG)
Made interneting difficult to maintain	(interneting,difficult,NEG)	(maintain,difficult,POS)	(maintain,difficult,POS)	(interneting,difficult,NEG)
It has so much more speed and the screen is very sharp	(speed,much more,POS) (screen,sharp,POS)	(screen,sharp,POS)	(screen,sharp,POS)	(screen,sharp,POS) (speed,more,POS)
The food was extremely tasty , creatively presented and the wine excellent	(food,tasty,POS) (food,creatively presented,POS) (wine,excellent,POS)	(food,tasty,POS) (food,creatively presented,POS) (wine,excellent,POS) (food,excellent,POS)	(food,tasty,POS) (food,creatively presented,POS) (wine,excellent,POS)	(food,tasty,POS) (food,creatively presented,POS) (wine,excellent,POS)

shows that S^3E^2 can better understand the context and distinguish the characteristics of words.

There are 2 triplets in the fourth example. All methods extract the triplet containing "screen". Unlike other models, S^3E^2 successfully identifies the second aspect "speed" and its sentiment. Though lacking of an opinion word "much", S^3E^2 has stronger recognition ability.

The last one is a more complicated example with 3 triplets, where an aspect corresponds to multiple opinions. We see that Grid-BiLSTM mistakenly matches "food" and "excellent" as a triplet. Both Grid-CNN and S^3E^2 make correct predictions. In general, the above analysis further proves that S^3E^2 can better understand the semantics and recognize the relationship more accurately.

5 Related Work

ASTE originates from another highly concerned research topic called *Aspect Based Sentiment Analysis* (ABSA) (Pontiki et al., 2014, 2015, 2016). The research process of ABSA can be divided into three stages.

Separate Extraction. Traditional studies have divided ABSA into three subtasks, namely, *aspect extraction* (AE), *opinion extraction* (OE), and *aspect sentiment classification* (ASC). The AE task (Yin et al., 2016; Li et al., 2018b; Xu et al., 2018; Ma et al., 2019) requires the extraction of aspects, while the OE task's goal (Fan et al., 2019) is to identify opinions expressed on them. The ASC task has attracted much more attention, which refers to classifying sentiment polarity for a given aspect target (Yang et al., 2017; Chen et al., 2017; Ma et al., 2018; Li et al., 2018a; Xue and Li, 2018; Wang et al., 2018; Li et al., 2019b) because the sentiment element carries crucial semantic information for a text. Zhang et al. (2019) develops *aspect-specific Graph Convolutional Networks* (ASGCN) that integrates with LSTM for the ASC task. Compared with ASGCN, S^3E^2 has richer edge types and fewer training parameters. Since its aspect-specific structure must depend on the given aspect, ASGCN lacks scalability and cannot be extended to triplet extraction in an end-to-end fashion. Besides, solving these three subtasks individually lacks practical application value and ignores the internal relation between them.

Pair Extraction. Recently, many studies have proposed effective models to jointly extract aspects and their sentiments (Zhang et al., 2015; Li and Lu, 2017, 2019; Li et al., 2019b,a). Hu et al. (2019) design a Span-Based method but conclude the pipeline model is better than the unified model. There is also a practice to co-extract aspects and opinions (Wang et al., 2017; Dai and Song, 2019). These pair extraction models still cannot fully understand a complete picture regarding sentiment and dig deeper into the interconnections between subtasks.

Triplet Extraction. The ASTE task is more challenging and application value. Peng et al. (2020) first propose a two-stage model for ASTE, which in the first stage co-extracts aspects with the associated sentiment and finishes opinion extraction in the form of a standard sequence labeling task. The second stage employs a binary classifier to match aspects and opinions to obtain final triplets. Following this work, Xu et al. (2020) employ a model with a position-aware tagging scheme to extract a triplet jointly, but it cannot apply to the one-to-many phenomenon. Wu et al. (2020) design a novel grid tagging schema to address triplet extraction, but their end-to-end model ignores the dependencies among words. Besides, the inference rounds of their inference strategy are not unified for each dataset, which may cause instability and high time complexity if the rounds rise.

6 Conclusion

Aspect Sentiment Triplet Extraction (ASTE) requires extracting aspects, corresponding opinions, and sentiment from user reviews. Different from previous work, we take advantage of multiple semantic relationships between word pairs and effectively capture the inner connection between such three elements. In this paper, we construct a novel model with a relational structure by creating a unique text graph for each sentence using *Graph Neural Network* (GNN). We also combine LSTM to obtain contextual semantics. Through the above mentioned rich structure, S^3E^2 can understand the context well and effectively recognize the identify between words. Besides, the inference strategy becomes more efficient because it only needs to be inferred once for all datasets, reducing the time complexity. Our end-to-end model achieves state-of-the-art performance on all datasets for triplet extraction. Experimental results show that S^3E^2 remarkably captures the connection between word pairs and recognizes their relationship.

Acknowledgments

This work is supported by the National Key Research and Development Program of China under Grant (No. 2020AAA0108501).

References

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5(0):135–146.

Peng Chen, Zhongqian Sun, Lidong Bing, and Wei Yang. 2017. Recurrent attention network on memory for aspect sentiment analysis. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 452–461.

Hongliang Dai and Yangqiu Song. 2019. Neural aspect and opinion term extraction with mined rules as weak supervision. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5268–5277.

Zhifang Fan, Zhen Wu, Xinyu Dai, Shujian Huang, and Jiajun Chen. 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, Volume 1 (Long and Short Papers)*, pages 2509–2518.

William L. Hamilton, Rex Ying, and Jure Leskovec. 2017. Inductive representation learning on large graphs. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, pages 1025–1035.

Ruidan He, Wee Sun Lee, Hwee Tou Ng, and Daniel Dahlmeier. 2019. An interactive multi-task learning network for end-to-end aspect-based sentiment analysis. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 504–515.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural Computation*, 9(8):1735–1780.

Minghao Hu, Yuxing Peng, Zhen Huang, Dongsheng Li, and Yiwei Lv. 2019. Open-domain targeted sentiment analysis via span-based extraction and classification. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 537–546.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *Proceedings of the 3rd International Conference on Learning Representations*.

Hao Li and Wei Lu. 2017. Learning latent sentiment scopes for entity-level sentiment analysis. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*, pages 3482–3489.

Hao Li and Wei Lu. 2019. Learning explicit and implicit structures for targeted sentiment analysis. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 5481–5491.

Xin Li, Lidong Bing, Wai Lam, and Bei Shi. 2018a. Transformation networks for target-oriented sentiment classification. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 946–956.

Xin Li, Lidong Bing, Piji Li, and Wai Lam. 2019a. A unified model for opinion target extraction and target sentiment prediction. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 6714–6721.

Xin Li, Lidong Bing, Piji Li, Wai Lam, and Zhimou Yang. 2018b. Aspect term extraction with history attention and selective transformation. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI 2018*, pages 4194–4200.

Zheng Li, Ying Wei, Yu Zhang, Xiang Zhang, and Xin Li. 2019b. Exploiting coarse-to-fine task transfer for aspect-level sentiment classification. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 4253–4260.

Bing Liu. 2012. Sentiment analysis and opinion mining. *Synthesis lectures on human language technologies*, 5(1):1–167.

Dehong Ma, Sujian Li, Fangzhao Wu, Xing Xie, and Houfeng Wang. 2019. Exploring sequence-to-sequence learning in aspect term extraction. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3538–3547.

Yukun Ma, Haiyun Peng, and Erik Cambria. 2018. Targeted aspect-based sentiment analysis via embedding commonsense knowledge into an attentive LSTM. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18)*, pages 5876–5883.

Haiyun Peng, Lu Xu, Lidong Bing, Fei Huang, Wei Lu, and Luo Si. 2020. Knowing what, how and why: A near complete solution for aspect-based sentiment analysis. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 8600–8607.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. GloVe: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543.

Maria Pontiki, Dimitris Galanis, Haris Papageorgiou, Ion Androutsopoulos, Suresh Manandhar, Mohammad AL-Smadi, Mahmoud Al-Ayyoub, Yanyan Zhao, Bing Qin, Orphée De Clercq, Véronique Hoste, Marianna Apidianaki, Xavier Tannier, Natalia Loukachevitch, Evgeniy Kotelnikov, Nuria Bel, Salud María Jiménez-Zafra, and Gülşen Eryiğit. 2016. SemEval-2016 task 5: Aspect based sentiment analysis. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pages 19–30.

Maria Pontiki, Dimitris Galanis, Haris Papageorgiou, Suresh Manandhar, and Ion Androutsopoulos. 2015. SemEval-2015 task 12: Aspect based sentiment analysis. In *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)*, pages 486–495.

Maria Pontiki, Dimitris Galanis, John Pavlopoulos, Harris Papageorgiou, Ion Androutsopoulos, and Suresh Manandhar. 2014. SemEval-2014 task 4: Aspect based sentiment analysis. In *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*, pages 27–35.

Peng Qi, Yuhao Zhang, Yuhui Zhang, Jason Bolton, and Christopher D. Manning. 2020. Stanza: A python natural language processing toolkit for many human languages. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 101–108.

Shuai Wang, Sahisnu Mazumder, Bing Liu, Mianwei Zhou, and Yi Chang. 2018. Target-sensitive memory networks for aspect sentiment classification. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 957–967.

Wenya Wang, Sinno Jialin Pan, Daniel Dahlmeier, and Xiaokui Xiao. 2017. Coupled multi-layer attentions for co-extraction of aspect and opinion terms. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*, pages 3316–3322.

Zhen Wu, Chengcan Ying, Fei Zhao, Zhifang Fan, Xinyu Dai, and Rui Xia. 2020. Grid tagging scheme for aspect-oriented fine-grained opinion extraction. In *Proceedings of Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 2576–2585.

Hu Xu, Bing Liu, Lei Shu, and Philip S. Yu. 2018. Double embeddings and CNN-based sequence labeling for aspect extraction. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, ACL 2018, Volume 2: Short Papers*, pages 592–598.

Lu Xu, Hao Li, Wei Lu, and Lidong Bing. 2020. Position-aware tagging for aspect sentiment triplet extraction. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*, pages 2339–2349.

Wei Xue and Tao Li. 2018. Aspect based sentiment analysis with gated convolutional networks. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2514–2523.

Min Yang, Wenting Tu, Jingxuan Wang, Fei Xu, and Xiaojun Chen. 2017. Attention-based LSTM for target-dependent sentiment classification. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*, pages 5013–5014.

Yichun Yin, Furu Wei, Li Dong, Kaimeng Xu, Ming Zhang, and Ming Zhou. 2016. Unsupervised word and dependency path embeddings for aspect term extraction. In *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence, IJCAI 2016*, pages 2979–2985.

Chen Zhang, Qiuchi Li, and Dawei Song. 2019. Aspect-based sentiment classification with aspect-specific graph convolutional networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4560–4570.

Meishan Zhang, Yue Zhang, and Duy-Tin Vo. 2015. Neural networks for open domain targeted sentiment. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 612–621.