Inheriting the Wisdom of Predecessors: A Multiplex Cascade Framework for Unified Aspect-based Sentiment Analysis

Hao Fei, **Fei Li**, **Chenliang Li**, **Shengqiong Wu**, **Jingye Li** and **Donghong Ji***

Key Laboratory of Aerospace Information Security and Trusted Computing, Ministry of Education, School of Cyber Science and Engineering, Wuhan University, China

{hao.fei, lifei_csnlp, cllee, whuwsq, theodorelee, dhji}@whu.edu.cn

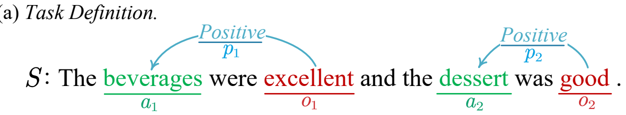
Abstract

So far, aspect-based sentiment analysis (ABSA) has involved with total seven subtasks, in which, however the interactions among them have been left unexplored sufficiently. This work presents a novel multiplex cascade framework for unified ABSA and maintaining such interactions. First, we model total seven subtasks as a hierarchical dependency in the easy-to-hard order, based on which we then propose a multiplex decoding mechanism, transferring the sentiment layouts and clues in lower tasks to upper ones. The multiplex strategy enables highly-efficient subtask interflows and avoids repetitive training; meanwhile it sufficiently utilizes the existing data without requiring any further annotation. Further, based on the characteristics of aspect-opinion term extraction and pairing, we enhance our multiplex framework by integrating POS tag and syntactic dependency information for term boundary and pairing identification. The proposed **Sy**ntax-aware **Mu**ltiple**x** (SyMux) framework enhances the ABSA performances on 28 subtasks (7×4 datasets) with big margins.

1 Introduction

As one of the core directions of sentiment analysis, ABSA has received extensive research attentions within past decade [Wang *et al.*, 2017; Li *et al.*, 2018; Fei *et al.*, 2021b]. ABSA has derived a number of subtasks, which all revolve around predicting three major elements of ABSA, i.e., **aspect**, **opinion** and **sentiment polarity**, or their combinations. For example, aspect term extraction (AE) and opinion term extraction (OE) seek to extract aspect and opinion terms respectively. Aspect-level sentiment classification (ALSC) aims at predicting the sentiment polarity given an aspect term, while triplet extraction (TE) targets as extracting all the correlated <aspect, opinion, polarity> triplets in a sentence. We illustrate the definitions of all the subtasks in Fig. 1(a) with specific examples. Most prior works solve one certain subtask in isolation [Wang *et al.*, 2017;

(a) *Task Definition.*



Subtask	Input→Output
Aspect Term Extraction(AE)	$S \to a_1, a_2$
Opinion Term Extraction(OE)	$S \to o_1, o_2$
Aspect-oriented Opinion Extraction(AOE)	$(S, a_1)/(S, a_2) \to o_1/o_2$
Aspect-opinion Pair Extraction(AOPE)	$S \to (a_1, o_1), (a_2, o_2)$
Aspect-level Sentiment Classification(ALSC)	$(S, a_1)/(S, a_2) \to p_1/p_2$
Aspect Extraction and Sentiment Classification(AESC)	$S \to (a_1, p_1), (a_2, p_2)$
Triplet Extraction(TE)	$S \to (a_1, o_1, p_1), (a_2, o_2, p_2)$

(b) *Subtask Correlations.*

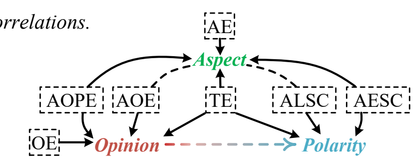


Figure 1: The definitions of all ABSA subtasks (a) and their correlations (b). In (b), the task names point to prediction targets with solid arrow lines. Dash lines indicate the tasks take the element as input.

Fan *et al.*, 2018], while some efforts have been made to construct unified methods for modeling as more subtasks as possible, so as to narrow the gaps between the modeling divergences of them [Yan *et al.*, 2021].

On the other hand, since different subtasks share much interactive correlation in essence, some works consider explicitly incorporating such underlying shared signals [Li *et al.*, 2019; Hu *et al.*, 2019]. Representatively, Chen and Qian [2020] and Yu *et al.* [2021] propose to collaboratively model the potential mutual interactions between ABSA subtasks based on the multi-task learning framework. Unfortunately, the interactions in these studies are limited to a small subset of ABSA subtasks (e.g., AE, OE and AESC), being inadequate enough to explore the rich shared information among all total seven subtasks. As depicted in Fig. 1(b), AE summarizes sentiment aspects, while OE emphasizes opinion words and meanwhile entails sentiment polarities. All the other subtasks are associated with each other based on dyadic (e.g., aspect-opinion) or triadic relations (e.g., aspect-opinion-polarity). Therefore, how to effectively and sufficiently capture the inline interactions among all these sub-

*Corresponding author

tasks for further enhancements still remains unexplored.

After carefully investigating the characteristics of the ABSA subtasks, we gain the following major observations:

▶ *Lower-level subtasks (e.g., AE and OE) actually lay the foundations of ABSA, explicitly offering the core sentiment layouts and clues for building up high-level subtasks.*

▶ *Higher-level subtasks (e.g., TE) contain more complete and richer information, which can in return benefit low-level tasks implicitly.*

▶ *All the subtasks can be unified by sharing the same 'aspect-opinion-polarity' ternary backbone, with slight divergences on inputs or outputs.*

Consequently, we consider modeling all these subtasks as a hierarchical dependency (HD) as detailed in Fig. 2, including four task groups ranging from Level A to Level D according to the task properties, i.e., from easy to hard.

On this basis, we then present a multiplex cascade framework for the unified ABSA. We formulate these subtasks with a new cascade grid-tagging scheme. As shown in Fig. 3, the system is built based on multi-task architecture, with all subtasks sharing one common encoder. Each decoder makes prediction for one certain subtask, all of which are cascaded following the HD structure, practicing the philosophy: **(1)** *rendering the ABSA keynotes with aspect and opinion (also entailing polarity) information produced at level A;* **(2)** *multiplexing the information of low-level tasks to the higher-level tasks one by one.* With such explicit multiplex design of reusing the knowledge from low-level subtasks, the predictions of upper subtasks will be gradually enhanced and the subtask interactions are tightly captured.

Further, we notice that the key bottlenecks of our multiplex framework lie in 1) the detection of the underlying aspect and opinion terms, i.e., OE and AE at level A, and 2) the pairing between aspect and opinion terms from level A to B. For further enhancement, we leverage the external syntax knowledge that has been extensively validated effective [Huang and Carley, 2019; Fei *et al.*, 2020]. First, the linguistic part-of-speech (POS) tags entail rich boundary information between span neighbors, which can essentially promote the recognition of aspect and opinion terms. Besides, the syntactic dependency features can provide additional clues for supporting more accurate reasoning of aspect-opinion pairing. We leverage a unified syntax graph convolution network [Wu *et al.*, 2021] to simultaneously model POS tags and dependency features at the encoding side. In the decoding side, we further introduce a syntax-guided pairing method by reharnessing the syntax weights yielded at the encoder.

Our framework takes the current existing ABSA datasets, where however not all training sentences has the annotations simultaneously covering all seven subtasks. We thus train seven decoders jointly on the shared training sets, while updating those low-level decoders with partial annotations. Evaluation results on 4 benchmark datasets suggest that the proposed system significantly outperforms all the current state-of-the-art ABSA models on total seven subtasks with big margins, demonstrating the effectiveness of our method for the unified ABSA. Our contributions include:

★ We build a unified model for total seven ABSA subtasks. We introduce a novel multiplex decoding based upon a cascade grid tagging method, fully reusing the shared information and interactions among different subtasks. Our model ensures high-efficient interflows, and meanwhile avoids duplicated or trivial training in previous single-task paradigm.

★ We propose to incorporate rich external syntactic knowledge, including POS tags and dependency trees, into both the encoder and decoders for enhancing aspect-opinion term extraction and pairing, which are the bottlenecks of all ABSA systems.

★ Our syntax-aware multiplex model (namely, SyMux) pushes the state-of-the-art results for all seven ABSA subtasks against best-performing baselines. The model can work without annotating any additional shared data, since the existing training data of high-level subtasks (e.g., TE) covers all the annotations needed by low-level subtasks.[1]

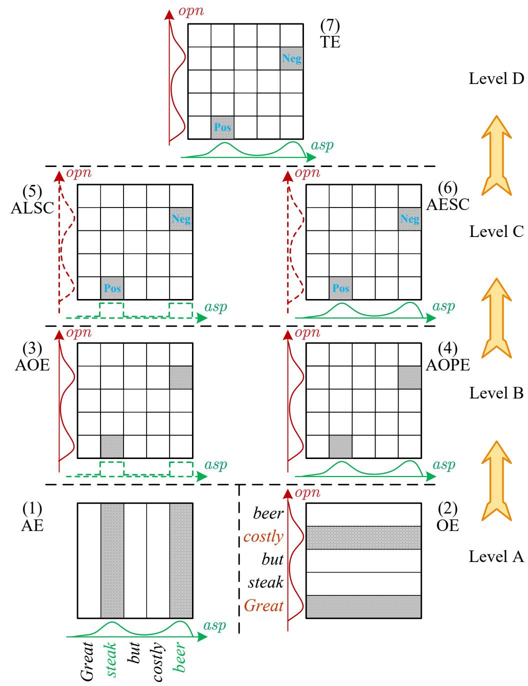


Figure 2: The hierarchical dependency (HD) for seven ABSA subtasks. ··· means the opinions are not prediction targets. ··· means the aspects are given as inputs. Lower-level subtasks predict basic information (e.g., aspects or opinions), while high-level subtasks incrementally predict more information (e.g., sentiment polarities) based on the information provided by lower-level subtasks.

2 Methodology

Task formulation. Based on the definition in Fig. 1(a), we now give the specific formulations of each subtask. The overall framework follows the multi-task learning scheme. The shared encoder takes an input sentence S=$\{w_1, \cdots, w_n\}$, and then seven separate decoders of different subtasks will yield their own outputs. For (1) AE, (2) OE, (3) AOE and (4) AOPE, the tag set is $\{\texttt{O},\texttt{I}\}$, where $\texttt{O}$ means a trivial word and $\texttt{I}$ means a word belonging to a term or forming an aspect-opinion pair. For (5) ALSC, (6) AESC and (7) TE, the tag

[1]Resources at https://github.com/scofield7419/UABSA-SyMux

set will be {O,P,N,M}, where O means a trivial word, and P/N/M imply that a word belongs to a term or an aspect-opinion pair meanwhile whose sentiment polarity is positive/negative/neutral, respectively. Specifically, for the base tasks OE and AE at level A, we take the 1-D sequential tagging scheme, while the high-level subtasks take the 2-D grid tagging [Wei *et al.*, 2020; Wu *et al.*, 2020].

Considering the example "*Great steak but costly beer*" in Figure 2, each decoder produces the following outputs Y:

(1) AE : $Y^1 = [\mathrm{O,I,O,O,I}]$, (2) OE : $Y^2 = [\mathrm{I,O,O,I,O}]$,

(3) AOE: $Y^3=\begin{bmatrix} \mathrm{O}&\mathrm{O}&\mathrm{O}&\mathrm{O}&\mathrm{O}\\ \mathrm{O}&\mathrm{O}&\mathrm{O}&\mathrm{O}&\underline{\mathrm{I}}\\ \mathrm{O}&\mathrm{O}&\mathrm{O}&\mathrm{O}&\mathrm{O}\\ \mathrm{O}&\mathrm{O}&\mathrm{O}&\mathrm{O}&\mathrm{O}\\ \mathrm{O}&\underline{\mathrm{I}}&\mathrm{O}&\mathrm{O}&\mathrm{O} \end{bmatrix}$, (4) AOPE: $Y^4=\begin{bmatrix} \mathrm{O}&\mathrm{O}&\mathrm{O}&\mathrm{O}&\mathrm{O}\\ \mathrm{O}&\mathrm{O}&\mathrm{O}&\mathrm{O}&\mathrm{I}\\ \mathrm{O}&\mathrm{O}&\mathrm{O}&\mathrm{O}&\mathrm{O}\\ \mathrm{O}&\mathrm{O}&\mathrm{O}&\mathrm{O}&\mathrm{O}\\ \mathrm{O}&\mathrm{I}&\mathrm{O}&\mathrm{O}&\mathrm{O} \end{bmatrix}$,

(5) ALSC: $Y^5=\begin{bmatrix} \mathrm{O}&\mathrm{O}&\mathrm{O}&\mathrm{O}&\mathrm{O}\\ \mathrm{O}&\mathrm{O}&\mathrm{O}&\mathrm{O}&\underline{\mathrm{N}}\\ \mathrm{O}&\mathrm{O}&\mathrm{O}&\mathrm{O}&\mathrm{O}\\ \mathrm{O}&\mathrm{O}&\mathrm{O}&\mathrm{O}&\mathrm{O}\\ \mathrm{O}&\underline{\mathrm{P}}&\mathrm{O}&\mathrm{O}&\mathrm{O} \end{bmatrix}$, (6) AESC: $Y^6=\begin{bmatrix} \mathrm{O}&\mathrm{O}&\mathrm{O}&\mathrm{O}&\mathrm{O}\\ \mathrm{O}&\mathrm{O}&\mathrm{O}&\mathrm{O}&\mathrm{N}\\ \mathrm{O}&\mathrm{O}&\mathrm{O}&\mathrm{O}&\mathrm{O}\\ \mathrm{O}&\mathrm{O}&\mathrm{O}&\mathrm{O}&\mathrm{O}\\ \mathrm{O}&\mathrm{P}&\mathrm{O}&\mathrm{O}&\mathrm{O} \end{bmatrix}$,

(7) TE: $Y^7=\begin{bmatrix} \mathrm{O}&\mathrm{O}&\mathrm{O}&\mathrm{O}&\mathrm{O}\\ \mathrm{O}&\mathrm{O}&\mathrm{O}&\mathrm{O}&\mathrm{N}\\ \mathrm{O}&\mathrm{O}&\mathrm{O}&\mathrm{O}&\mathrm{O}\\ \mathrm{O}&\mathrm{O}&\mathrm{O}&\mathrm{O}&\mathrm{O}\\ \mathrm{O}&\mathrm{P}&\mathrm{O}&\mathrm{O}&\mathrm{O} \end{bmatrix}$.

Note that underlined tags mean that gold-standard aspect terms are given, while those tags with vertical dots at left mean that opinions are not the required prediction targets.

2.1 Encoder

Contextual encoder. Pre-trained language models (PLM), e.g., BERT [Devlin *et al.*, 2019] have been shown prominent on retrieving the contextualized features, becoming the de-facto encoder in a wide range of NLP tasks [Eberts and Ulges, 2020; Zhao *et al.*, 2020]. In this work we take the BERT-variant PLM, RoBERTa, as the context encoder, in which the calculations are summarized as:

$$\{\boldsymbol{h}_1, \cdots, \boldsymbol{h}_n\} = \mathrm{RoBERTa}(\{w_1, \cdots, w_n\})\,, \tag{1}$$

where $\boldsymbol{h}_n$ is an output representation for word w_n.

Unified syntax GCN encoder. Essentially, the recognition of aspect and opinion terms as well as their pairing are two key foundations of our system. Therefore, we integrate external syntactic knowledge into our system for enhancement, including the word-level linguistic POS tags for potential span boundaries of aspect and opinion terms [Wu *et al.*, 2021], and the syntactic dependency trees for aspect-opinion pairing [Fei *et al.*, 2021a; Fei *et al.*, 2021c]. Following Wu *et al.* [2021], we leverage a unified syntax GCN (**USGCN**) encoder to fuse these two sources simultaneously.

Based on the input S we first have a list of POS tags $\{w_i^p\}_n$ for each word. We maintain the POS embedding vector $\boldsymbol{x}_i^p$ via a look-up table. Then we have the corresponding dependency tree of S. We form a graph $G = (V, E)$, where V is a set of words, and E is a set of dependency edges between each pair of words. We define an adjacency matrix $\{a_{i,j}\}_{n\times n}$ in E, in which $a_{i,j}$=1 if there is an edge between w_i and w_j, and $a_{i,j}$=0 vice versa. We additionally enable the self-loop of each word and bidirectional edges between valid word pair

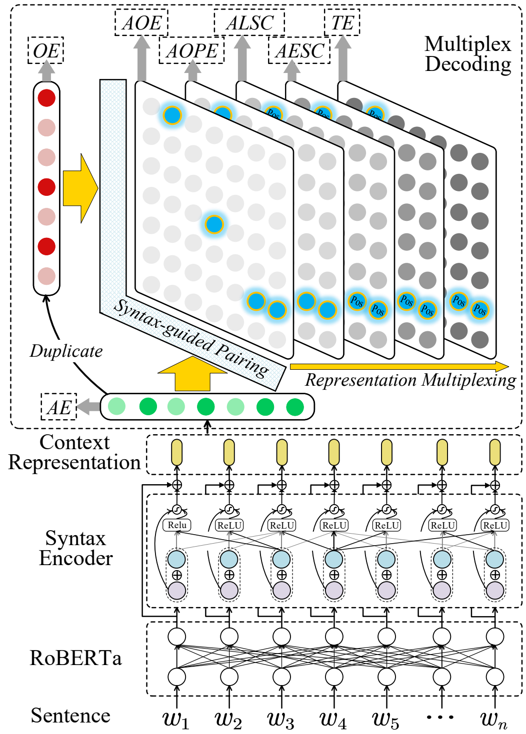


Figure 3: An overview of the **SyMux** framework. At encoding side a RoBERTa LM and a syntax GCN encoder sequentially generate the context representations for the input. At the decoding side, the AE and OE decoders first produce the predictions of aspect and opinion terms. Then, aspect and opinion representations are paired into a 2-D sentiment representation, which is further guided by the syntax information. Thereafter, the rest upper subtask decoders incrementally multiplex sentiment representations for predictions one by one.

for information enhancement. Suppose $\boldsymbol{e}_i^l$ is the hidden representation of w_i at l-th USGCN layer:

$$\boldsymbol{e}_i^l = \mathrm{ReLU}(\textstyle\sum_{j=1}^n \alpha_{i,j}^l (\boldsymbol{W}_1 \cdot [\boldsymbol{e}_j^{l-1}; \boldsymbol{x}_j^p]))\,, \tag{2}$$

where $\alpha_{i,j}^l$ is the syntactic-aware neighboring weight obtained from:

$$\alpha_{i,j}^l = \frac{a_{i,j}\cdot \exp\left(\boldsymbol{e}_i^{l-1} \otimes \boldsymbol{e}_j^{l-1}\right)}{\sum_{j=1}^n a_{i,j}\cdot \exp\left(\boldsymbol{e}_i^{l-1} \otimes \boldsymbol{e}_j^{l-1}\right)}\,, \tag{3}$$

where $\otimes$ is an element-wise multiplication. The weight matrix $\alpha_{i,j}$ entails rich syntactic relationship between tokens.

We then apply a gate c to flexibly coordinate the contribution between POS information and USGCN representations, since term recognition may rely directly on the span boundary signals provided by POS tags. We will empirically validate the usefulness of such operation in the experiments.

$$\begin{aligned} c_i^l &= \mathrm{Sigmoid}(\boldsymbol{W}_2[\boldsymbol{e}_i^l; \boldsymbol{x}_i^p])\,, \\ \boldsymbol{v}_i^{s,l} &= c_i^l \odot \boldsymbol{e}_i^l + (1 - c_i^l) \odot \boldsymbol{x}_i^p\,, \end{aligned} \tag{4}$$

where $\boldsymbol{v}_i^{s,l}$ is the final output of the l-th USGCN. We take total L layers of USGCN for full syntax knowledge propagation. Finally, we explicitly concatenate the outputs of USGCN and RoBERTa encoders as the final contextualized representation

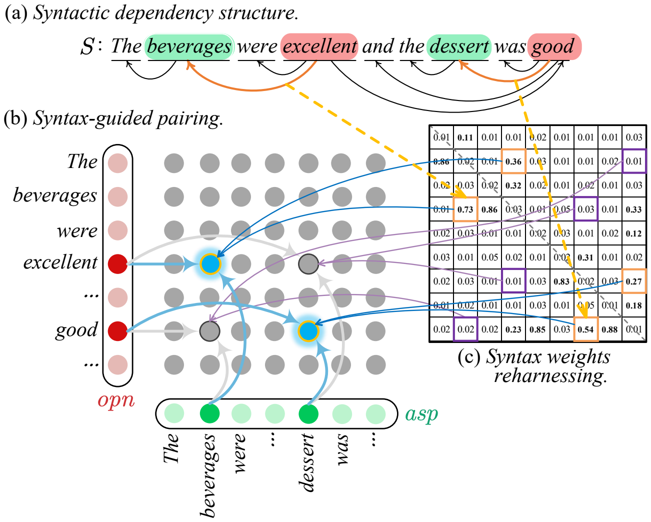


Figure 4: Syntax-guided aspect-opinion pairing. (b) reflects Eq. (6), and (c) is obtained from Eq. (3).

for each word: $\boldsymbol{v}_i = [\boldsymbol{v}_i^{s,L}; \boldsymbol{h}_i]$.

2.2 Decoder

AE and OE decoding. As shown in Fig. 3, the AE decoder first obtains its representation $\boldsymbol{R}^1$=$\{\boldsymbol{r}_i^1\}_n$ based on the context representation $\boldsymbol{v}_i$ via a non-linear transformation, i.e., $\boldsymbol{r}_i^1$=FFNs$(\boldsymbol{v}_i)$. Also the representation $\boldsymbol{R}^1$ will be duplicated as the OE decoder representation $\boldsymbol{R}^2$=$\{\boldsymbol{r}_i^2\}_n$ via another non-linear transformation. AE and OE decoders then produce their own predictions (i.e., Y^1, Y^2) based on their 1-D representations with two separate softmax classifiers.

Syntax-guided pairing. The next step is to construct a 2-D representation $\boldsymbol{R}^M$=$\{\boldsymbol{r}_{i,j}^M\}_{n\times n}$ based on $\boldsymbol{R}^1$ and $\boldsymbol{R}^2$ for the aspect-opinion pairs. We reach this via Cartesian product:

$$\boldsymbol{r}_{i,j}^M = \boldsymbol{W}_3[\boldsymbol{r}_i^1; \boldsymbol{r}_j^2]\,, \tag{5}$$

where the vertical direction in $\boldsymbol{R}^M$ denotes aspect terms while the horizontal direction denotes opinion terms. To aid the pairing, we further utilize the syntactic dependency signals. Intuitively, the syntax structure reveals how aspect and opinion terms relate to each others, as depicted in Fig. 4(a). Although we encode the dependency features previously via USGCN, the information can be quite diluted at the decoding stage. We thus reharness the syntax weights calculated in Eq. (3) for the pairing, i.e., syntax-guided pairing, as in Fig. 4.

$$\boldsymbol{r}_{i,j}^M = \frac{1}{2}(\alpha_{i,j}^L + \alpha_{j,i}^L)\cdot \boldsymbol{W}_3[\boldsymbol{r}_i^1; \boldsymbol{r}_j^2]\,. \tag{6}$$

Multiplex decoding. As we formulate the unified ABSA subtasks into HD (cf. Fig. 2), all the remaining subtasks can share one same sentiment layout in our multiplex framework. Naturally, almost all the pivotal information needed for detecting the downstream subtasks is entailed in the initial 2-D representation $\boldsymbol{R}^M$. Thus we perform representation multiplexing based on $\boldsymbol{R}^M$ along the order of [1/2→(3→4)→(5→6)→7] within the HD structure, as can be seen in Fig. 3.

Technically, we obtain the decoder representation $\boldsymbol{R}^k$ ($3\leq k\leq 7$) of the forward k-th subtask via the chain operation $\boldsymbol{R}^k \leftarrow \boldsymbol{R}^{k-1}$:

$$\boldsymbol{R}^{k,S} = \boldsymbol{R}^{k-1}\cdot \boldsymbol{W}_4^{k,S}\,, \tag{7}$$

$$\boldsymbol{R}^k = \boldsymbol{g}^k \odot \boldsymbol{R}^{k,S} + (1-\boldsymbol{g}^k)\odot \boldsymbol{R}^{k,P}\,, \tag{8}$$

where $\boldsymbol{R}^{k,P}$=$\{\boldsymbol{r}_{i,j}^{k,P}\}_{n\times n}$ is a private trainable 2-D parameter of k-th subtask for maintaining the task-private features, $\boldsymbol{W}_4^{k,S} \in \mathbb{R}^{n\times n}$ is a trainable transition matrix, and $\boldsymbol{g}^k$ is a gate controlling the strength between shared and private information. Note that $\boldsymbol{R}^2$ is initialized with $\boldsymbol{R}^M$.

For AOPE, AESC and TE, the private representation $\boldsymbol{R}^{k,P}$ is randomly initialized. For AOE and ALSC where aspect annotations are available, we particularly inject the gold aspect signals into $\boldsymbol{R}^{k,P}$ by concatenating an embedding $\boldsymbol{x}_i^a$ which is a binary aspect indicator, where 1 denotes an aspect term and vice versa for 0.

$$\boldsymbol{r}_{i,j}^{3,P} \leftarrow [\boldsymbol{r}_{i,j}^{3,P}; \boldsymbol{x}_i^a]\,, \tag{9}$$

$$\boldsymbol{r}_{i,j}^{5,P} \leftarrow [\boldsymbol{r}_{i,j}^{5,P}; \boldsymbol{x}_i^a]\,, \tag{10}$$

where '←' represents an affine transformation.

Predicting. Based on the representations $\boldsymbol{R}^k$, each subtask yields its own prediction Y^k via a softmax classifier on each elements of the 2-d matrix. For ALSC and AESC, the opinion terms in such 2-D prediction are not required, and only the aspects or polarities will be output. It is also worth noticing that in prior work some subtasks including AOE, ALSC and AESC are generally modeled as 1-D formulation (i.e., sequential labeling or sentence classification) since their inputs or outputs relate mainly to one certain element (i.e., either aspects or opinions/polarities). But in our multiplex framework, both two elements are fully considered as complementary sentiment clues for further enhancements.

2.3 Training

We perform multi-task training of our framework, where the input sentences should come with annotations of whole total seven subtasks. However, not all the training sentences has been labeled simultaneously with all subtasks. Therefore, we on the one hand perform error back-propagation for all subtask decoders on those shared training sets; on the other hand we partially train those decoders with input sentences that have target annotations and meanwhile keep the rest decoders fixed. The objective is to jointly optimize all seven subtasks:

$$\mathcal{L} = \textstyle\sum_{k=1}^{7}\lambda_k \mathcal{L}_k\,, \tag{11}$$

where λ_k ($\sum_{k=1}^{7}\lambda_k = 1$) is a co-efficiency of k-th task loss:

$$\mathcal{L}_k = -\frac{1}{N}\sum \hat{Y}^k \log Y^k\,, \tag{12}$$

here N is the number of training set for task k, and $\hat{Y}^k$ is the ground-truth labels.

3 Experiments

3.1 Setups

Following Yan et al. (2021), we use the Semeval benchmark [Pontiki *et al.*, 2014; Pontiki *et al.*, 2015; Pontiki *et al.*, 2016], including Res14, Lap14, Res15 and Res16, which

		Sent.	AE/ALSC /AESC	OE	AOE/AOPE	TE
Res14	Source	/	D_{17}&D_{20}	D_{17}&D_{20}	D_{19}&D_{20}	D_{20}
	Train(Shrd.)	1,497	2,112	2,439	2,439	2,439
	Train(Prvt.)	1,663	2,160	2,011	518	230
	Test	800	1,226	1,289	1,000	862
Lap14	Source	/	D_{17}&D_{20}	D_{17}&D_{20}	D_{19}&D_{20}	D_{20}
	Train(Shrd.)	1,035	1,242	1,420	1,420	1,420
	Train(Prvt.)	2,112	2,443	2,442	413	182
	Test	800	988	1,079	572	490
Res15	Source	/	D_{17}&D_{20}	D_{17}&D_{20}	D_{19}&D_{20}	D_{20}
	Train(Shrd.)	665	867	1,014	1,014	1,014
	Train(Prvt.)	724	851	865	330	147
	Test	685	850	905	532	455
Res16	Source	/	D_{20}	D_{20}	D_{19}&D_{20}	D_{20}
	Train(Shrd.)	1,052	1,380	1,605	1,605	1,605
	Train(Prvt.)	25	0	0	33	0
	Test	320	405	465	476	465

Table 1: Data statistics. 'Shrd.': shared annotations over all tasks. 'Prvt.': private annotations for certain tasks.

have only the annotations for AE, ALSC and AESC. Follow-up works further partially contribute labels for different subtasks based on the Semeval sentences. Representatively, Wang *et al.* [2017] annotate the unpaired opinion terms (denoted as D_{17}), while Fan *et al.* [2019] pair the aspects with opinion terms (D_{19}), and Peng *et al.* [2020] provide the labels for triple extraction (D_{20}). To enable multi-task training, we re-ensemble them so that most of the sentences' annotations cover all seven subtasks. Table 1 shows the data statistics.

We use the base version of RoBERTa PLM. L=3 in US-GCN. Most of the representations have the shape of 300-d. The loss co-efficiency λ_k is fine-tuned for different tasks.

We make comparisons against some top-performing ABSA baselines, following Yan et al. (2021), including the models for ABSA subtasks at level A and C, such as SPAN [Hu *et al.*, 2019], IMN [He *et al.*, 2019], RACL [Chen and Qian, 2020] and MIN [Yu *et al.*, 2021], and the systems capable of addressing all subtasks: CMLA [Wang *et al.*, 2017], RINANTE [Dai and Song, 2019], Li-unified [Li *et al.*, 2019], GTS [Wu *et al.*, 2020], Dual-MRC [Mao *et al.*, 2021], GEN [Yan *et al.*, 2021]. All the comparing models install PLM (e.g., BERT, BART) for fair comparisons. For each subtask, we adopt the same evaluation metrics as the corresponding previous works do. We report the F1 scores for all tasks.

3.2 Results and Analyses

Main results. Table 2 shows the overall results on four datasets. We see that the unified systems (such as Dual-MRC and GEN) can bring very strong performances, while those models (e.g., RACL and MIN) considering rich interactions among subtasks can even achieve higher results on several certain subtasks. Yet our SyMux system significantly outperforms all these best-performing baselines universally on all subtasks over all datasets with big margins. This demonstrates the effectiveness of our method. More interestingly, we can observe that the higher-level tasks can benefit more from our system, comparing to those tasks at bottom levels, e.g., AE and oE at level A. Especially, ALSC, AESC and TE tasks always gain over 2 points of F1 over four datasets, compared with the best baseline. This substantially implies the necessity to take the multiplex idea for unified ABSA, fully reusing and recycling of the information learned at the lower level for facilitating the upper level tasks. Also it is worth explicitly noticing that such improvements of all subtasks by our system are obtained via the multiplex mechanism, without using any additional ABSA annotations.

	Lvl-A		Lvl-B		Lvl-C		Lvl-D
	AE	OE	AOE	AOPE	ALSC	AESC	TE
• *Res14*							
SPAN	86.71	84.36	/	/	71.75	73.68	/
IMN	84.06	85.10	/	/	75.67	70.72	/
RACL	86.38	87.18	/	/	81.61	75.42	/
MIN	87.91	85.66	/	/	80.48	76.02	/
CMLA	81.22	83.07	79.53	48.95	78.65	70.62	43.12
RINANTE	81.34	83.33	80.68	46.29	76.18	48.15	34.03
Li-unified	81.62	85.26	81.36	55.34	74.55	73.79	51.68
GTS	83.82	85.04	82.04	75.53	80.22	74.85	70.20
Dual-MRC	86.60	86.22	83.73	74.93	82.04	76.57	70.32
GEN	87.07	87.29	85.38	77.68	75.56	73.56	72.46
SyMux	**89.02**	**88.54**	**87.05**	**79.42**	**84.45**	**78.68**	**74.84**
Δ	+1.11	+1.25	+1.67	+1.74	+2.41	+2.09	+2.38
• *Lap14*							
SPAN	82.34	79.58	/	/	62.50	61.25	/
IMN	77.55	81.00	/	/	75.56	61.73	/
RACL	81.79	79.72	/	/	73.91	63.40	/
MIN	83.22	81.80	/	/	74.95	64.83	/
CMLA	79.53	78.68	77.95	44.10	70.20	56.90	32.90
RINANTE	80.40	77.13	75.34	29.70	68.45	36.70	20.00
Li-unified	78.56	78.54	77.55	52.56	70.03	63.38	42.47
GTS	82.48	79.52	78.61	65.67	73.85	64.23	54.58
Dual-MRC	82.51	80.44	79.90	63.37	75.97	65.94	55.58
GEN	83.52	77.86	80.55	66.11	76.76	68.17	57.59
SyMux	**84.42**	**82.55**	**81.97**	**67.64**	**78.99**	**70.32**	**60.11**
Δ	+0.90	+0.75	+1.42	+1.53	+2.23	+2.15	+2.52
• *Res15*							
SPAN	74.63	76.85	/	/	50.28	62.29	/
IMN	69.90	73.29	/	/	70.10	60.22	/
RACL	73.99	76.00	/	/	74.91	66.05	/
CMLA	76.03	74.67	73.42	44.60	71.50	53.60	35.90
RINANTE	73.38	75.40	72.50	35.40	71.33	41.30	28.00
Li-unified	74.65	74.25	75.32	56.85	70.64	64.95	46.69
GTS	78.22	79.31	76.41	67.53	72.67	65.30	58.67
Dual-MRC	75.08	77.52	74.50	64.97	73.59	65.08	57.21
GEN	75.48	76.49	80.52	67.98	73.91	66.61	60.11
SyMux	**79.73**	**80.71**	**82.42**	**69.82**	**77.51**	**69.08**	**63.13**
Δ	+1.51	+1.40	+1.71	+1.84	+2.60	+2.47	+3.02
• *Res16*							
SPAN	74.68	72.45	/	/	82.23	82.23	/
RACL	74.91	73.56	/	/	81.36	68.58	/
CMLA	74.20	72.20	80.63	50.00	78.32	61.20	41.60
RINANTE	72.82	70.45	79.34	30.70	75.13	42.10	23.30
Li-unified	73.36	73.87	80.66	53.75	79.60	71.20	44.51
GTS	75.80	76.38	82.79	74.62	83.21	70.68	67.58
Dual-MRC	76.87	77.90	83.33	75.71	84.68	70.84	67.40
GEN	81.35	80.54	87.92	77.38	86.20	75.69	69.98
SyMux	**82.41**	**81.68**	**89.88**	**78.82**	**88.62**	**77.95**	**72.76**
Δ	+1.06	+1.14	+1.96	+1.44	+2.42	+2.26	+2.78

Table 2: Performances (F1) on each subtasks. Δ means our improvements over the second-best ones (underlined). Values in ▢ are copied from Yan et al. (2021), in ▢ are from Wu et al. (2020), in ▢ are from Yu et al. (2021), in ▢ are from our re-implementation (average results over five runs). ▢ means that the method is not applicable for that subtask.

Ablation. Since we take the external syntax information for

	Lvl-A		Lvl-B		Lvl-C		Lvl-D
	AE	OE	AOE	AOPE	ALSC	AESC	TE
Dual-MRC	86.60	86.22	83.73	74.93	82.04	76.57	70.32
GEN	87.07	87.29	85.38	77.68	75.56	73.56	72.46
SyMux	**89.02**	**88.54**	**87.05**	**79.42**	**84.45**	**78.68**	**74.84**
w/o POS	87.01	86.27	86.53	78.41	84.04	77.80	74.19
Δ	-2.01	-2.27	-0.52	-1.01	-0.41	-0.88	-0.65
w/o Gate	87.97	87.32	86.73	78.95	84.33	78.46	74.75
Δ	-1.05	-1.22	-0.32	-0.47	-0.12	-0.22	-0.09
w/o SGP	88.87	88.40	86.13	77.84	83.67	77.42	73.88
Δ	-0.15	-0.14	-0.92	-1.58	-0.78	-1.26	-0.96

Table 3: Ablation results on *Res14*. The most significant ablation items are highlighted in .

	Lvl-A		Lvl-B		Lvl-C		Lvl-D
	AE	OE	AOE	AOPE	ALSC	AESC	TE
• *Separating*							
Dual-MRC	86.60	86.22	83.73	74.93	82.04	76.57	70.32
GEN	87.07	87.29	85.38	77.68	75.56	73.56	72.46
• *Partial Sharing*							
IMN	84.06	85.10	/	/	75.67	70.72	/
RACL	86.38	87.18	/	/	81.61	75.42	/
MIN	87.91	85.66	/	/	80.48	76.02	/
• *Partial Sharing + Multiplex*							
	87.78	87.53	/	/	/	/	/
	88.59	88.21	86.75	79.18	/	/	/
	88.32	87.95	/	/	83.87	78.02	/
SyMux	88.67	88.28	86.86	79.30	/	/	74.10
	88.72	88.34	/	/	84.18	78.23	74.45
	88.97	88.49	86.97	79.34	84.29	78.34	/
• *Full Sharing + Multiplex*							
SyMux	**89.02**	**88.54**	**87.05**	**79.42**	**84.45**	**78.68**	**74.84**

Table 4: Results (on *Res14*) with different sharing mechanisms among subtasks. The tasks with results marked as '/' in sharing-aware methods are made unshareable.

enhancement, here we investigate the impacts. By removing the POS tag features away from USGCN, as in Table 3, there can be notable performance drops on AE and OE. This proves the importance of the linguistic POS knowledge for helping the term boundary detection. Likewise, if uninstalling the POS gate c (cf. Eq. 4) the overall results are also hurt, especially for the level-A tasks. This mostly verifies our assumption that it should have more sufficient and direct access to boundary information (i.e., POS tags) to aid the aspect or opinion extraction. Further stripping off the dependency knowledge from decoding, i.e., without syntax-guided pairing (SGP), we witness big performance reductions on all the higher-level 2-D subtasks. On the contrary, this influences very little on the AE and OE tasks. This further verifies that the POS and dependency features contribute to unified ABSA from different angles. And making use of both two features is complementarily favorable for the tasks.

Separating or sharing? sharing by multiplex! Recent related research tries to model the interactions between ABSA subtasks via sharing-aware learning, which has not been considered by those separating-learning methods. In Table 4,

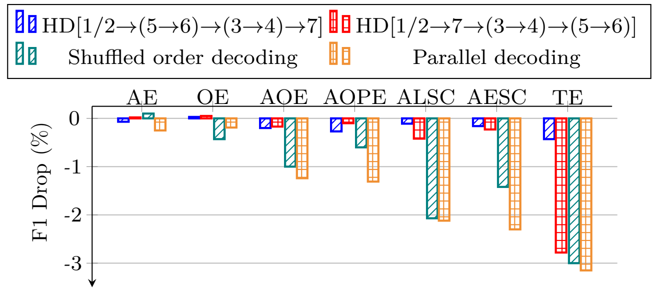


Figure 5: Performance drops using other decoding orders.

we show the comparisons between '*Separating*' and '*Partial Sharing*', and we learn that those baselines considering sharing mechanisms (e.g., RACL, MIN) can actually be beaten by those recent separating-learning models that employ certain powerful strategies, e.g., taking stronger pre-trained language model of BART in GEN, or remodeling the tasks as reading comprehension in DUAL-MRC. However, our sharing-based framework can surpass all strong separating-learning models. Via this, we prove that it is still important to perform sharing-aware learning for unified ABSA. Notedly, with the multiplex mechanism, our system with partial sharing can still substantially outperform the separating-learning methods. For example, by merely sharing the four tasks at level A and B, ours greatly outperforms GEN and DUAL-MRC on all subtasks. In the meantime, this implies that our multiplex system can be of great usefulness in real-world applications, i.e., applied to some more strict scenarios where not all the data for total of seven subtasks are available.

As we revealed earlier, those existing sharing-aware baselines trying to promote the interactions among subtasks are unfortunately limited to a very small subset of tasks, i.e., partial sharing. Here we can observe that the full-scale information sharing of our model (in '*Full Sharing+Multiplex*') enables to produce more accurate predictions, providing most sufficient interactions, comparing to any partial sharing scheme. Further looking into the SyMux itself, different schemes of partial sharing lead to varied performances, and more sharing generally gives better results. Furthermore, even with the same partial sharing as in baselines (i.e., shared on level A and C), ours still performs much better. We thus can conclude that the key to a successful sharing-aware learning among ABSA subtasks is the multiplex method.

Multiplexing order in hierarchical dependency. We suggest a hierarchical dependency of subtasks with the order of [1/2→(3→4)→(5→6)→7], which ensures a reasonable information interflow from low to high (level A to level D). Here we explore the impacts when following other different multiplexing orders. In Fig. 5 we plot the performance drop based on *Res14*. Overall, using other decoding orders can hurt all tasks' performances. Especially, the shuffled order and the parallel order, which are essentially the equivalents to getting rid of multi-task learning based multiplex strategy, lead to the biggest decreases. Even exchanging the subtasks at level B and level C can result in slight drops.

Model robustness against data scarcity. Owing to the mul-

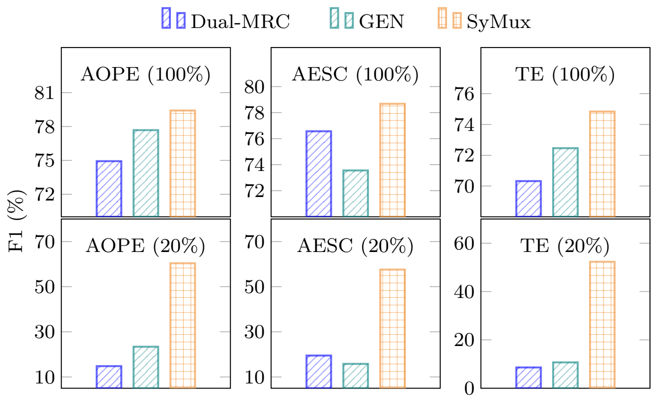


Figure 6: ABSA results on data-rich (100%) and data-scarce (20%) setting where we only reduce the annotations for the target subtask, while the labels for training other tasks are available in full scale.

tiplex decoding design, the follow-up tasks in the HD structure can greatly benefit from their predecessor tasks. Here we further study the performances of each subtask when the annotations for the corresponding task are scarce, i.e., few-shot learning. Fig. 6 shows the results of three representative high-level tasks (AOPE, AESC and TE) based on *Res14* data. On the data-rich setting, i.e., with sufficient training signals, each system performs normally (ours performs the best). However, when only feeding 20% training annotations for training each of the three tasks respectively, the performances drop substantially. By comparison, our system even without using the full annotations for training one certain task, obtains much better results than baselines. This directly confirms that the multiplex cascade framework can fully recycle the signals of prior subtasks, e.g., sentiment layouts or clues, which semi-supervisedly supports the reasoning of the topper tasks.

4 Related Work

The studies of ABSA mainly center on three key factors, **aspect**, **opinion** and **sentiment polarity**, which together depict the complete sentiment picture, i.e., aspect indicates *what* the sentiment targets are, sentiment polarity describes *how* the sentiment strengths are, and opinion explains *why* have such polarities. Within the scope of the ABSA, so far there can be a set of specific subtasks, each of them focuses on one specific element or the combinations of them, including 1) aspect term extraction (**AE**) [Li and Lam, 2017], 2) opinion term extraction (**OE**) [Wang *et al.*, 2017], 3) aspect-oriented opinion extraction (**AOE**) [Fan *et al.*, 2019], 4) aspect-opinion pair extraction (**AOPE**) [Zhao *et al.*, 2020], 5) aspect-level sentiment classification (**ALSC**) [Tang *et al.*, 2016], 6) aspect extraction and sentiment classification (**AESC**) [Hu *et al.*, 2019] and 7) triplet extraction (**TE**) [Peng *et al.*, 2020].

At the initial stage, separate solutions are extensively adopted for those standalone subtasks (e.g., AE, OE, ALSC) [Li and Lam, 2017; Fan *et al.*, 2018]. Later, hybrid ABSA subtasks (e.g., AOE, AESC, TE) have been put on and solved with joint models [Li *et al.*, 2019; Peng *et al.*, 2020]. And then, some unified methods have been proposed to model as many subtasks as possible, so as to narrow the gaps between the modeling divergences of them [Chen *et al.*, 2020; Mao *et al.*, 2021]. Very recently, Yan *et al.* [2021] unify total seven subtasks with a pointer-network based sequence-to-sequence framework. Unlike them, in this work, we construct a unified ABSA model to facilitate the information exchange. On the other hand, several studies attempt to promote the interactions between some ABSA subtasks. For example, Chen and Qian [2020] and Yu *et al.* [2021] take the multi-task strategy to enhance the collaborative learning between the tasks of AE, OE and AESC. Yet these works only consider a very small subset of the subtasks. In this work, we reveal that actually the underlying information of all subtasks can largely be reused, based on which we propose a novel syntax-aware multiplex framework for better unification.

5 Conclusions

We investigate a syntax-aware multiplex method for unified ABSA. We model total seven ABSA subtasks as a hierarchical dependency, based on which we present a multiplex cascade decoding framework, fully reusing the shared sentiment information and layout among different tasks. On the other hand, we incorporate both syntactic POS tag and dependency tree features for enhancing aspect-opinion term extraction and pairing. Our framework enables high-efficient interflows of ABSA subtasks, and meanwhile avoids duplicated and trivial training in previous separate-learning methods. Experimental results show that our method outperforms many state-of-the-art baselines for ABSA. In-depth analyses reveal the system's strengths, e.g., data robustness. We also find that our model performs well when the annotated data of some subtasks are not available or the data size is quite small, which suggests that the method has a wider potential application scenarios of ABSA. Future directions can revolve around the data-driven attempts for solving unbalanced sentiment polarities and inconsistent annotations of subtasks.

Acknowledgments

This work is supported by the National Natural Science Foundation of China (No.61772378, No. 62176187), the National Key Research and Development Program of China (No. 2017YFC1200500), the Research Foundation of Ministry of Education of China (No.18JZD015), the Youth Fund for Humanities and Social Science Research of Ministry of Education of China (No. 22YJCZH064), the General Project of Natural Science Foundation of Hubei Province (No.2021CFB385). This work is also supported by the Fundamental Research Funds for the Central Universities.

References

[Chen and Qian, 2020] Zhuang Chen and Tieyun Qian. Relation-aware collaborative learning for unified aspect-based sentiment analysis. In *ACL*, pages 3685–3694, 2020.

[Chen *et al.*, 2020] Peng Chen, Shaowei Chen, and Jie Liu. Hierarchical sequence labeling model for aspect sentiment triplet extraction. In *NLPCC*, pages 654–666, 2020.

[Dai and Song, 2019] Hongliang Dai and Yangqiu Song. Neural aspect and opinion term extraction with mined rules as weak supervision. In *ACL*, pages 5268–5277, 2019.

[Devlin *et al.*, 2019] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *NAACL*, pages 4171–4186, 2019.

[Eberts and Ulges, 2020] Markus Eberts and Adrian Ulges. Span-based joint entity and relation extraction with transformer pre-training. In *ECAI*, pages 2006–2013, 2020.

[Fan *et al.*, 2018] Feifan Fan, Yansong Feng, and Dongyan Zhao. Multi-grained attention network for aspect-level sentiment classification. In *EMNLP*, pages 3433–3442, 2018.

[Fan *et al.*, 2019] Zhifang Fan, Zhen Wu, Xin-Yu Dai, Shujian Huang, and Jiajun Chen. Target-oriented opinion words extraction with target-fused neural sequence labeling. In *NAACL*, pages 2509–2518, 2019.

[Fei *et al.*, 2020] Hao Fei, Yafeng Ren, and Donghong Ji. Retrofitting structure-aware transformer language model for end tasks. In *EMNLP*, pages 2151–2161, 2020.

[Fei *et al.*, 2021a] Hao Fei, Fei Li, Bobo Li, and Donghong Ji. Encoder-decoder based unified semantic role labeling with label-aware syntax. In *AAAI*, pages 12794–12802, 2021.

[Fei *et al.*, 2021b] Hao Fei, Yafeng Ren, Shengqiong Wu, Bobo Li, and Donghong Ji. Latent target-opinion as prior for document-level sentiment classification: A variational approach from fine-grained perspective. In *Proceedings of the WWW: the Web Conference*, pages 553–564, 2021.

[Fei *et al.*, 2021c] Hao Fei, Shengqiong Wu, Yafeng Ren, Fei Li, and Donghong Ji. Better combine them together! integrating syntactic constituency and dependency representations for semantic role labeling. In *Findings of the Association for Computational Linguistics: ACL/IJCNLP 2021*, pages 549–559, 2021.

[He *et al.*, 2019] Ruidan He, Wee Sun Lee, Hwee Tou Ng, and Daniel Dahlmeier. An interactive multi-task learning network for end-to-end aspect-based sentiment analysis. In *ACL*, pages 504–515, 2019.

[Hu *et al.*, 2019] Minghao Hu, Yuxing Peng, Zhen Huang, Dongsheng Li, and Yiwei Lv. Open-domain targeted sentiment analysis via span-based extraction and classification. In *ACL*, pages 537–546, 2019.

[Huang and Carley, 2019] Binxuan Huang and Kathleen Carley. Syntax-aware aspect level sentiment classification with graph attention networks. In *EMNLP*, pages 5469–5477, 2019.

[Li and Lam, 2017] Xin Li and Wai Lam. Deep multi-task learning for aspect term extraction with memory interaction. In *EMNLP*, pages 2886–2892, 2017.

[Li *et al.*, 2018] Xin Li, Lidong Bing, Wai Lam, and Bei Shi. Transformation networks for target-oriented sentiment classification. In *ACL*, pages 946–956, 2018.

[Li *et al.*, 2019] Xin Li, Lidong Bing, Piji Li, and Wai Lam. A unified model for opinion target extraction and target sentiment prediction. In *AAAI*, pages 6714–6721, 2019.

[Mao *et al.*, 2021] Yue Mao, Yi Shen, Chao Yu, and Longjun Cai. A joint training dual-mrc framework for aspect based sentiment analysis. In *AAAI*, pages 13543–13551, 2021.

[Peng *et al.*, 2020] Haiyun Peng, Lu Xu, Lidong Bing, Fei Huang, Wei Lu, and Luo Si. Knowing what, how and why: A near complete solution for aspect-based sentiment analysis. In *AAAI*, pages 8600–8607, 2020.

[Pontiki *et al.*, 2014] Maria Pontiki, Dimitris Galanis, John Pavlopoulos, Harris Papageorgiou, Ion Androutsopoulos, and Suresh Manandhar. Semeval-2014 task 4: Aspect based sentiment analysis. In *SemEval*, pages 27–35, 2014.

[Pontiki *et al.*, 2015] Maria Pontiki, Dimitrios Galanis, Suresh Manandhar, and Ion Androutsopoulos. Semeval-2015 task 12: Aspect based sentiment analysis. In *SemEval*, pages 486–495, 2015.

[Pontiki *et al.*, 2016] Maria Pontiki, Salud María Jiménez-Zafra, and Gülşen Eryiğit. SemEval-2016 task 5: Aspect based sentiment analysis. In *SemEval*, pages 19–30, 2016.

[Tang *et al.*, 2016] Duyu Tang, Bing Qin, Xiaocheng Feng, and Ting Liu. Effective LSTMs for target-dependent sentiment classification. In *COLING*, pages 3298–3307, 2016.

[Wang *et al.*, 2017] Wenya Wang, Sinno Jialin Pan, Daniel Dahlmeier, and Xiaokui Xiao. Coupled multi-layer attentions for co-extraction of aspect and opinion terms. In *AAAI*, pages 3316–3322, 2017.

[Wei *et al.*, 2020] Zhepei Wei, Jianlin Su, Yue Wang, Yuan Tian, and Yi Chang. A novel cascade binary tagging framework for relational triple extraction. In *ACL*, pages 1476–1488, 2020.

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

[Wu *et al.*, 2021] Shengqiong Wu, Hao Fei, Yafeng Ren, Donghong Ji, and Jingye Li. Learn from syntax: Improving pair-wise aspect and opinion terms extraction with rich syntactic knowledge. In *IJCAI*, pages 3957–3963, 2021.

[Yan *et al.*, 2021] Hang Yan, Junqi Dai, Tuo Ji, Xipeng Qiu, and Zheng Zhang. A unified generative framework for aspect-based sentiment analysis. In *ACL*, pages 2416–2429, 2021.

[Yu *et al.*, 2021] Guoxin Yu, Xiang Ao, Ling Luo, Min Yang, Xiaofei Sun, Jiwei Li, and Qing He. Making flexible use of sub-tasks: A multiplex interaction network for unified aspect-based sentiment analysis. In *Findings of the ACL*, pages 2695–2705, 2021.

[Zhao *et al.*, 2020] He Zhao, Longtao Huang, Rong Zhang, Quan Lu, and Hui Xue. Spanmlt: A span-based multi-task learning framework for pair-wise aspect and opinion terms extraction. In *ACL*, pages 3239–3248, 2020.