SK2: Integrating Implicit Sentiment Knowledge and Explicit Syntax Knowledge for Aspect-Based Sentiment Analysis

Jia Li*
Peking University
Beijing, China
lijiaa@pku.edu.cn

Yuyuan Zhao*
Peking University
Beijing, China
zhaoyuyuan@pku.edu.cn

Zhi Jin†
Peking University
Beijing, China
zhijin@pku.edu.cn

Ge Li†
Peking University
Beijing, China
lige@pku.edu.cn

Tao Shen
UTS
Sydney, Australia
tao.shen@uts.edu.au

Zhengwei Tao
Peking University
Beijing, China
tttzw@stu.pku.edu.cn

Chongyang Tao
Peking University
Beijing, China
chongyangtao@gmail.com

ABSTRACT

Aspect-based sentiment analysis (ABSA) plays an indispensable role in web mining and retrieval system as it involves a wide range of tasks, including aspect term extraction, opinion term extraction, aspect sentiment classification, etc. Early works are merely applicable to a part of these tasks, leading to computation-unfriendly models and a pipeline framework. Recently, a unified framework has been proposed to learn all these ABSA tasks in an end-to-end fashion. Despite its versatility, its performance is still sub-optimal since ABSA tasks depend heavily on both sentiment and syntax knowledge, but existing task-specific knowledge integration methods are hardly applicable to such a unified framework. Therefore, we propose a brand-new unified framework for ABSA in this work, which incorporates both implicit sentiment knowledge and explicit syntax knowledge to better complete all ABSA tasks. To effectively incorporate implicit sentiment knowledge, we first design a self-supervised pre-training procedure that is general enough to all ABSA tasks. It consists of conjunctive words prediction (CWP) task, sentiment-word polarity prediction (SPP) task, attribute nouns prediction (ANP) task, and sentiment-oriented masked language modeling (SMLM) task. Empowered by the pre-training procedure, our framework acquires strong abilities in sentiment representation and sentiment understanding. Meantime, considering a subtle syntax variation can significantly affect ABSA, we further explore a sparse relational graph attention network (SR-GAT) to introduce explicit aspect-oriented syntax knowledge. By combining both worlds of knowledge, our unified model can better represent and understand the input texts towards all ABSA tasks. Extensive experiments show that our proposed framework achieves consistent and significant improvements on all ABSA tasks.

KEYWORDS

Aspect-based sentiment analysis, information extraction, pre-trained language model, deep neural network

ACM Reference Format:
Jia Li, Yuyuan Zhao, Zhi Jin, Ge Li, Tao Shen, Zhengwei Tao, and Chongyang Tao. 2022. SK2: Integrating Implicit Sentiment Knowledge and Explicit Syntax Knowledge for Aspect-Based Sentiment Analysis. In *Proceedings of the 31st ACM International Conference on Information and Knowledge Management (CIKM '22), October 17–21, 2022, Atlanta, GA, USA.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3511808.3557452

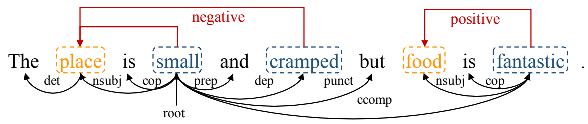
1 INTRODUCTION

Aspect-based sentiment analysis (ABSA) aims to address a wide range of sentiment analysis tasks at a fine-grained level. Compared to sentence-level sentiment analysis [20, 23] that focuses merely on overall sentiment polarity, ABSA breaks a sentence into *aspect terms* and *opinion terms*, and then discriminate the sentiment polarity towards each aspect by considering its corresponding opinion term (i.e., *aspect sentiment classification*). As an example shown in Figure 1(top), in the sentence "the place is small and cramped but food is fantastic", the aspect terms are "place" and "food", their sentiment polarities are "negative" and "positive", and the opinion terms are "small", "cramped", and "fantastic". As ABSA captures every single perspective towards intra-sentence sentiments, it plays an indispensable role in web mining for downstream applications [1, 13], e.g., analysis of drug reviews [9] and understanding hospitality service [28], and has attracted increasing attention in recent years.

*Both authors contributed equally to this research. Jia Li, Zhi Jin, Ge Li, and Zhengwei Tao are with the Key Lab of High Confidence Software Technologies (PKU), Ministry of Education, China.
†Corresponding author.

CIKM '22, October 17–21, 2022, Atlanta, GA, USA

ACM ISBN 978-1-4503-9236-5/22/10…$15.00
https://doi.org/10.1145/3511808.3557452



Target	Subtask	Output
Extract Terms	AE	place, food
	OE	small, cramped, fantastic
	AOE	small, cramped / fantastic
	AOPE	(place, small), (place, cramped), (food, fantastic)
Predict Sentiments	ASC	negative given place / positive given food
Compound	AESC	(place, negative), (food, positive)
	ASTE	(place, small, negative), (place, cramped, negative), (food, fantastic, positive)

Figure 1: Examples of aspect terms, opinion terms, and their sentiment polarities (top), and seven ABSA tasks (bottom).

In detail, ABSA involves seven tasks for intra-sentence fine-grained sentiment analysis. Besides the three fundamental tasks, i.e., aspect extraction (AE), opinion extraction (OE), and aspect sentiment classification (ASC), it also involves their combinations as four extra tasks. That is, aspect extraction and sentiment classification (AESC), aspect-based opinion extraction (AOE), aspect and opinion pair extraction (AOPE), and aspect sentiment triplet extraction (ASTE). Given the above example, these seven tasks' targets are shown in Figure 1(bottom). In the past few years, researchers have paid tremendous efforts to improve the performance of these tasks [30, 42] through utilizing advanced representation methods [34, 46], exploring new architectures [2, 13], or incorporating various task-specific knowledge [29, 44]. Although some works achieve impressive performance on a particular task, they ignore the intrinsic connection among those ABSA tasks. Besides, these works handle each task individually or a part of them, leading to a pipeline framework and computation-unfriendly models.

Very recently, Mao et al. [21] formatted the ABSA tasks as two machine reading comprehension problems and jointly learned all tasks with a shared pre-trained language model (PLM). Xu et al. [43] exploited a sequence-to-sequence model to solve all ABSA tasks. Despite their versatility, such a straightforward multi-task fine-tuning manner can only achieve sub-optimal results due to the lack of sentiment-aware knowledge and aspect-oriented syntax knowledge: Firstly, different from some coarse-grained NLP tasks, ABSA tasks depend more on fine-grained sentiment-aware words and prior information including conjunctive words, attribute nouns, sentiment words, and their sentiment polarities. Despite the power of PLM in learning generic semantic representations, the fine-grained sentiment-aware knowledge is seldom explicitly studied and exploited in the PLM, thus it is challenging and sub-optimal if we directly fine-tune the pre-trained language models on the ABSA tasks. Secondly, although some probing works have shown that PLM is equipped with implicit syntax knowledge (e.g., word-pair dependency), ABSA tasks rely more on aspect-related syntax knowledge, where the syntax tree roots on an aspect term and entails direct interactions between aspect and opinion terms [34]. The aspect-related syntax can remarkably improve the neural model to explicitly capture the sentiment towards the aspect.

Motivated by above, we propose a brand-new unified framework integrated with both implicit Sentiment Knowledge and explicit Syntax Knowledge (thus dubbed as SK2) for improving performance on all ABSA tasks. Specifically, we design a self-supervised pre-training procedure to effectively introduce fine-grained *implicit sentiment knowledge*. The self-supervised pre-training procedure has four tasks – conjunctive words prediction (CWP) task, sentiment-word polarity prediction (SPP) task, attribute nouns prediction (ANP) task, and sentiment-oriented masked language modeling (SMLM) task – encouraging PLMs to be sentiment-aware. In contrast to previous task-specific sentiment pre-training [31], we keep the pre-training tasks general enough to benefit all downstream ABSA tasks. Empowered by the general pre-training procedure, our unified model can acquire more strong abilities in sentiment representation and sentiment understanding. In addition, we explore a sparse relational graph attention network (SR-GAT) to precisely introduce *explicit aspect-oriented syntax knowledge* to PLMs in fine-tuning phase. To be specific, instead of directly using a dependency parsing tree that has many unnecessary edges between words to interfere with modeling relations between aspects and opinion, we propose a sparse aspect-oriented syntactic tree. It begins with a dependency parsing tree given a sentence and transforms the tree into a sparse and aspect-rooted syntactic tree. Then, we apply the SR-GAT to encode the syntactic tree structure and incorporate the syntax encodings of graph into Transformer. Consequently, by combining both worlds of knowledge, our unified model can better represent and understand the input texts towards all ABSA tasks.

We evaluate the proposed SK2 on all ABSA tasks including aspect extraction (AE), opinion extraction (OE), aspect sentiment classification (ASC), aspect extraction and sentiment classification (AESC), aspect-based opinion extraction (AOE), aspect and opinion pair extraction (AOPE), and aspect sentiment triplet extraction (ASTE). Experimental results show that our SK2 can consistently improve the performance over all metrics on three benchmarks. Specifically, on the AE, OE, and ASC tasks, our SK2 brings 2.89%, 1.87%, and 5.43% average absolute improvements on F1 score respectively; On the AESC, AOE, and AOPE tasks, SK2 achieves 1.74%, 2.73%, and 2.28% average absolute improvements on F1, respectively. Meanwhile, our SK2 achieves 1.97% average absolute improvements on the ASTE task which is the hardest among all ABSA tasks.

In summary, our main contributions are four folds: (i) We propose a brand-new unified framework that integrates both implicit sentiment knowledge and explicit syntax knowledge to benefit all ABSA tasks. (ii) To incorporate implicit sentiment knowledge, we introduce four tasks in the self-supervised pre-training procedure, which can effectively contribute to sentiment representation and sentiment understanding. (iii) We also propose an explicit method to integrate syntax knowledge by our sparse aspect-oriented tree and SR-GAT, which can be regarded as a plug-in module to benefit broad ABSA works. (iv) Experimental results verify that our framework can achieve consistent and significant improvements on all seven ABSA tasks in three benchmarks.

2 PROPOSED FRAMEWORK

Figure 2 gives an overview of our unified framework. We consider building the unified framework based on PLMs as it is pre-trained

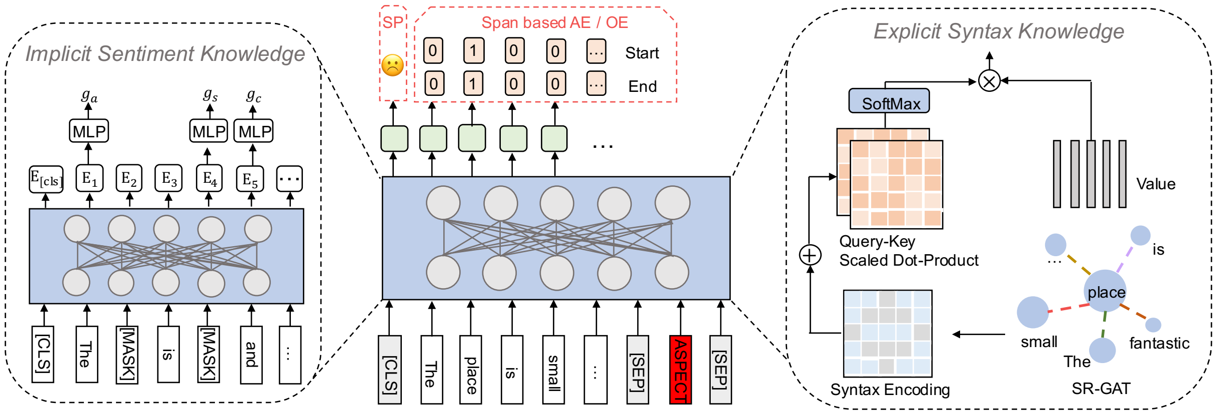


Figure 2: The framework of our proposed SK2.

on a large amount of unlabeled data and shows strong ability in language representation and language understanding. Without loss of generality, we choose BERT [6] as the backbone. To effectively incorporate fine-grained implicit sentiment knowledge, we design a self-supervised pre-training procedure including the CWP, SPP, ANP, and SMLM tasks. Based on this, we further explore a sparse relational graph attention network (SR-GAT) to introduce explicit aspect-oriented syntax knowledge in the multi-task fine-tuning phase. The SR-GAT can help model to capture the relationship between the aspect terms and opinion terms as much as possible. By combining both worlds of knowledge, our SK2 acquires strong abilities in sentiment representation and sentiment understanding, and can better handle the input texts towards all ABSA tasks.

Formally, given an input sentence $s = \{w_1, w_2, ..., w_n\}$ where n is the number of tokens, SK2 produces all aspect terms $\{a_1, a_2, ...a_q\}$ for AE task, all opinion terms $\{o_1, o_2, ...o_j\}$ for OE task, all opinions $\{o_1, o_2, ...o_k\}$ based on a specific aspect for AOE task, all aspect and opinion pairs $\{(a_1, o_1), ...(a_z, o_z)\}$ for AOPE task, and sentiment polarities of specific aspects $\{s_1, ...s_v\}$ for ASC task, all aspect terms and their corresponding sentiments $\{(a_1, s_1), ...(a_t, s_t)\}$ for AESC task, and a set of triples $\{(a_1, o_1, s_1), ...(a_m, o_m, s_m)\}$ for ASTE tasks. $a_k = \{w_i, ..., w_j\}$ is the k-th aspect term in the sentence s, which is a single word or a phrase. o_k means the k-th opinion term in the sentence s. Likewise, o_k is a single word or a phrase. s_k represents the sentiment polarity of a_k where $s_k \in \{\texttt{Positive}, \texttt{Neutral}, \texttt{Negative}\}$.

Next, we will describe the process of integrating implicit sentiment knowledge and explicit syntax knowledge into SK2, the details of multi-task fine-tuning, and how SK2 effectively handles all ABSA tasks with the unified model.

2.1 Integrating Implicit Sentiment Knowledge

Despite the power of PLMs in learning generic semantic representations, the fine-grained sentiment-aware knowledge is seldom explicitly exploited in the PLMs, thus it is sub-optimal if we directly fine-tune the pre-trained language models on the ABSA tasks. To address the above problem, we design a self-supervised pre-training procedure to introduce fine-grained implicit sentiment knowledge for BERT. Concretely, the pre-training procedure includes conjunctive words prediction, sentiment-word polarity prediction, attribute nouns prediction, and sentiment-oriented masked language model. In the next, we will introduce them in detail.

Table 1: Selected representative conjunctive words.

$\text{CWP}_{y_c=1}$	and, also, besides, additional, furthermore, too, still, moreover, in addition, like wise, as well, what's more, not only…but also..., as well as, $\cdots$
$\text{CWP}_{y_c=0}$	but, however, whereas, though, nevertheless, instead, conversely, in contrast, instead of, but unless, even though, by contrast, on the contrary, $\cdots$

2.1.1 *Conjunctive Words Prediction (CWP).*

The CWP task is introduced to teach BERT when the sentiment polarity of a sentence changes. Considering that the sentiment polarity of a sentence usually keeps the same or is the opposite after conjunctive words, we design the novel task to predict the polarity of conjunctive words. As shown in Figure 1, after the conjunctive word "but", the sentiment polarity changes from "negative" to "positive". Specifically, we first select one hundred representative conjunctive words and divide them into two categories as shown in Table 1. The first category represents a progressive relationship labeled with $y_c = 1$, where the sentiment polarity remains unchanged. The other category indicates a transitional relationship labeled with $y_c = 0$, and the sentiment polarity changes after it. Given a token sequence $\{w_1, w_2, ..., w_n\}$ where w_t is a conjunctive token, we feed the sequence into BERT and obtain a representation sequence $\{E_1, E_2, ..., E_n\}$. Then the model predicts the polarities of w_t on its output representation E_t. The conjunctive score $g_c(t)$ can be calculated by a non-linear transformation. Finally, the objective function of CWP task is formulated as:

$$\mathcal{L}_{cwp} = -\frac{1}{n_c}\sum_{t=1}^{n_c} y_c^t log(g_c(t)) \tag{1}$$

where y_c^t is the label of the t-th conjunctive token and n_c is the number of conjunctive tokens in the sequence.

2.1.2 *Attribute Nouns Prediction (ANP).*

Since ABSA tasks are mainly applied to analyze reviews, e.g., laptop reviews and restaurant reviews, there exist a large number of attribute nouns. Table 2 shows the attribute nouns of different-domain reviews that reviewers have expressed opinions on. Therefore, recognizing attribute nouns is crucial for extracting aspect terms. To inject attribute nouns

Table 2: Exhibition of attribute nouns on restaurant domain and laptop domain. Bold indicates attribute nouns and Underline means common nouns.

	Restaurant Domain
Example 1	**Pizza** and **garlic** knots are great as well , I order from them quite often and the **delivery** is always super quick.
Example 2	Although the tables may be closely situated , **food** quality and **service** overcompensate.
	Laptop Domain
Example 1	I love the solid machined **aluminum** frame, and the **keyboard** is the best of any **laptop** I 've used.
Example 2	I thought the white **Mac computers** looked dirty too quickly where you use the **mouse** and where you place your hands when typing.

knowledge, we introduce the ANP task. We first employ the TF-IDF algorithm to identify attribute nouns by comparing the occurrence frequency of the noun n in the domain-specific texts [1] with the occurrence frequency of the noun n in the open domain texts.

After acquiring the attribute nouns, we train BERT with the ANP task. Given a token sequence $\{w_1, w_2, ..., w_n\}$, we mask all the attribute nouns with [MASK] token. The model is required to predict the masked tokens based on the rest tokens. Then we feed the masked sequence into BERT and obtain a representation sequence $\{E_1, E_2, ..., E_n\}$. Finally, given that the l-th token is masked, the probability distribution $g_a(l)$ of E_l over vocabulary can be calculated by a non-linear transformation. The training objective of ANP task is to minimize the following negative log-likelihood (NLL):

$$\mathcal{L}_{anp} = -\frac{1}{n_a}\sum_{l=1}^{n_a} y_a^l log(g_a(l)) \quad (2)$$

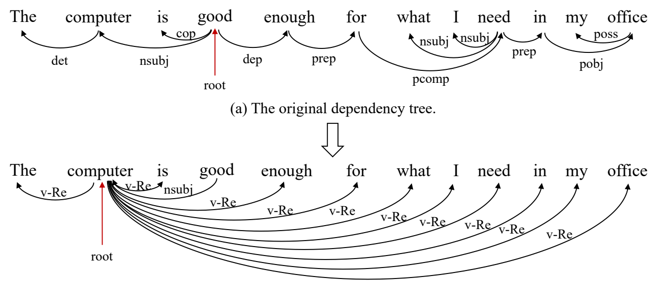
where y_a^l is the label of the masked l-th token and n_a is the number of masked tokens in the sequence.

2.1.3 Sentiment-word Polarity Prediction (SPP). As described in ANP task, ABSA tasks are mainly used for reviews which are mostly emotional sentences containing a great many sentiment words. Therefore, inspired by Tian et al. [31], we introduce the SPP task to inject sentiment-word polarity knowledge and provide weak supervision for the model to predict sentiment polarities. We first apply Pointwise Mutual Information (PMI) algorithm [32] for sentiment knowledge mining. Specifically, we structure a seed collection $\mathcal{R}$ that contains fifty representative sentiment-word seeds and their corresponding sentiment polarities. The seed collection $\mathcal{R}$ is represented as $\mathcal{R} = \{(x_1, s_1), ..., (x_m, s_m)\}$ where x_i is a sentiment-word seed and s_i is its sentiment polarity. Then we build a seed-candidate collection $Q = \{(x_1, c_1), ..., (x_n, c_n)\}$ where c_i is a sentiment-word candidate, which keeps the same part-of-speech pattern as Turney [32]. The PMI between seed word x_k and candidate word c_k is calculated as:

$$\mathtt{PMI}(x_k, c_k) = log\frac{p(x_k, c_k)}{p(x_k)p(c_k)} \quad (3)$$

where $p(x_k, c_k)$ is the co-occurring probability of x_k and c_k in all sentences. The sentiment polarity of c_k is determined by the difference between its PMI scores on all positive seeds and all negative

[1] Domain-specific texts are about laptop and restaurant reviews which have the same domains with our pre-training corpus.



(a) The original dependency tree.

(b) The sparse aspect-oriented dependency tree.

Figure 3: A sparse aspect-oriented dependency tree (b) constructed from an original dependency tree (a).

seeds, which is formulated as:

$$\mathtt{SP}(c_k) = \sum_{\mathtt{SP}(x_j)=+}\mathtt{PMI}(x_j, c_k) - \sum_{\mathtt{SP}(x_j)=-}\mathtt{PMI}(x_j, c_k) \quad (4)$$

If $\mathtt{SP}(c_k)$ is larger than 0, we regard c_k as a positive sentiment word labeled with $y_p^k = 1$, otherwise $y_p^k = 0$. Then we feed a sequence $\{w_1, ..., c_k, ..., w_n\}$ into BERT and acquire a representation sequence $\{E_1, ..., E_k, ..., E_n\}$ where c_k is a sentiment token. The polarity score $g_p(k)$ of c_k is predicted by a non-linear transformer on E_k. The training objective $\mathcal{L}_{spp}$ of SPP task is defined as:

$$\mathcal{L}_{spp} = -\frac{1}{n_m}\sum_{k=1}^{n_m} y_p^k log(g_p(k)) \quad (5)$$

where n_m is the number of sentiment tokens in the sequence.

2.1.4 Sentiment-oriented Masked Language Modeling (SMLM). As one of the common self-supervised tasks in PLMs, token-level masked language modeling is usually utilized to guide the model to learn semantic features of word sequences with the bidirectional context. To capture sentiment representations, we introduce the SMLM task, which can guide the model to learn sentiment features of input sequences. Specifically, we replace sentiment tokens which can be acquired as described in the SPP task with [MASK] token. If the number of the masked attribute tokens and sentiment tokens is less than 15% of the input sequence, we randomly select other tokens to mask. SMLM task is modeled as a multi-classification problem for each masked token and the objective $\mathcal{L}_{smlm}$ is the same with Formula 2.

2.1.5 Joint Learning. We take a multi-task learning manner to optimize the model, and the final loss in self-supervised pre-training procedure is the sum of losses from the CWP, ANP, SPP, and SMLM tasks. The final objective function is formulated as:

$$\mathcal{L}_{pt} = \mathcal{L}_{cwp} + \mathcal{L}_{anp} + \mathcal{L}_{spp} + \mathcal{L}_{smlm} \quad (6)$$

Based on the self-supervised pre-training procedure, SK2 is incorporated implicit sentiment knowledge and becomes more sentiment awareness, which is beneficial for aspect term extraction, opinion term extraction, and sentiment polarity prediction.

2.2 Integrating Explicit Syntax Knowledge

ABSA tasks rely more on aspect-related syntax knowledge, where the syntax tree roots on an aspect term and entails direct interactions between aspect and opinion terms. The aspect-related syntax can prompt the neural model to explicitly capture the sentiment towards the aspect. Inspired by this point, we further explore a sparse relational graph attention network (SR-GAT) to incorporate explicit aspect-oriented syntax knowledge for the subsequent ABSA tasks, as illustrated in the right part of Figure 2.

The traditional dependency tree contains abundant grammar information and many are usually not rooted at a target aspect term, as indicated in Figure 3. Nevertheless, the focus of ABSA tasks is the target aspect terms instead of the root of dependency tree. Moreover, in the dependency tree, some dependency relationships are not essential and may introduce noise since the key of ABSA tasks is the dependency relations between the aspect terms and the opinion terms. Therefore, we propose a sparse aspect-oriented dependency tree by reshaping and pruning the original dependency tree. Concretely, inspired by Wang et al. [34], we first reshape the dependency tree into an aspect-oriented dependency tree. If the sentence with multiple aspect terms, we construct a unique tree for each target separately. Different from Wang et al. [34], considering that some dependency relationships are not essential, we prune the tree to reduce the impact of other unrelated tokens and noisy relations. Specifically, we acquire the dependency relations that are directly connected to the target aspect and only retain six dependency relations {`amod, dobj, neg, nsubj, rcmd, xcomp`}. According to the definition of dependency relationships in De Marneffe and Manning [5], the selected six relations can grasp the essential clues to extract opinion terms for the specific aspect term. For example, as shown in Figure 3(b) , the "nsubj" relation between "computer" and "good" provides effective clues for ABSA tasks, but the "det" relation between "computer" and "the" offers minimal information in Figure 3(a). Next, we treat other dependency relations as virtual relations (v-Re) from the target aspect to each corresponding node.

Based on this, we explore a sparse relational graph attention network (SR-GAT) to represent the sparse aspect-oriented dependency tree. In SR-GAT $\mathcal{G}$, each node denotes peer token of the input sequence, edge represents the dependency relation between two nodes. The neighborhood nodes of node n_i are $\mathcal{N}_i$. We first initialize node representations with the summation of the token embedding and POS embedding. Formally, the i-th node representation of the $(k+1)$-th layer can be computed as:

$$\begin{aligned} h_i^{k+1} &= ||_{m=1}^{M} \sum_{j \in \mathcal{N}_i} \alpha i j^{km} W_m^k h_j^k \\ g_{ij}^{km} &= \sigma(\texttt{relu}(r_{ij}^{km} W_{m1} + b_{m1}) W_{m2} + b_{m2}) \\ \alpha i j^{km} &= \frac{\exp(g_{ij}^{km})}{\sum_{j=1}^{\mathcal{N}_i} \exp(g_{ij}^{km})} \end{aligned} \tag{7}$$

where M is the number of relational heads. h_j^k is the representation of node j in the k-th layer. W_{m1}, W_{m2}, W_m^k, b_{m1} and b_{m2} are trainable parameters. r_{ij}^{km} is the dependency relationship embedding between node i and j in the m-th head of the k-th layer. The above aggregation operation is performed K times and the final dependency relationship features of the m head is $R^{Km} \in \mathbb{R}^{n \times n \times d_p}$ where n is the length of input sequence and d_p is the dimension of dependency relationship embeddings.

Since dependency relationship features contain sufficient syntax information towards the target aspect, we propose to incorporate these syntax knowledge into Transformer. Concretely, Transformer architecture consists of a composition of Transformer layers [33]. Each Transformer layer has a self-attention module and a position-wise feed-forward network (FFN). The input of each layer is $H \in \mathbb{R}^{n \times d}$ where d is the hidden dimension. H is then projected into Q, K, and V through three matrices $W_Q \in \mathbb{R}^{d \times d_K}$, $W_K \in \mathbb{R}^{d \times d_K}$, and $W_V \in \mathbb{R}^{d \times d_V}$. To incorporate syntax knowledge, we compute an average of the dot-products of the dependency relationship features and a learning parameters in all heads. Then we introduce the dependency features via a bias term to the attention module, and revise the formulation of the self-attention mechanism as:

$$\begin{aligned} A &= softmax\left(\frac{(HW_Q)(HW_K)^T}{\sqrt{d_K}} + S\right)V \\ S &= \frac{1}{M} \sum_{m=1}^{M} R^{Km} W_S^T \end{aligned} \tag{8}$$

where $W_S \in \mathbb{R}^{dp}$ is a learning parameter.

Until now, our BERT has possessed sentiment knowledge and syntax knowledge through the proposed self-supervised pre-training procedure and incorporating dependency features from the SR-GAT. With the help of both procedures, BERT has strong abilities in sentiment representation and sentiment understanding.

2.3 Multi-task Fine-tuning

Based on integrating implicit sentiment knowledge and explicit syntax knowledge, SK2 can effectively handle all ABSA tasks in a unified framework. In the next, we will describe the multi-task fine-tuning process of SK2 in detail, including aspect term extraction, opinion term extraction, and sentiment polarity prediction.

2.3.1 Extract Aspect Terms. To extract aspects, we construct the following input for BERT: `[CLS] Sentence [SEP] Aspect [SEP]`, where `Aspect` is the word "aspect", and `[CLS]` and `[SEP]` are special tokens for classification and separating sentences. Meanwhile, we feed the sparse aspect-oriented dependency tree into SR-GAT[2] and integrate syntax features into BERT as described in Equation 8. Then BERT outputs the final representations $\tilde{h}_a^K$ of input. Instead of using a sequence tagging strategy, we apply a span-based scheme [12] to determine aspect terms, which predicts the start and end positions of targets under the supervision of target span in the input. Next, we apply two non-linear transformations based on $\tilde{h}_a^K$ to calculate the start position score $g_{a,s}^i$ and end position score $g_{a,e}^i$ of the the i-th aspect term. Finally, we extract all aspect terms from the input by minimizing the negative log-likelihood (NLL):

$$\mathcal{L}_a = -\sum_{i=1}^{q} y_{a,s}^i log g_{a,s}^i - \sum_{i=1}^{q} y_{a,e}^i log g_{a,e}^i \tag{9}$$

where $y_{a,s}^i$ and $y_{a,e}^i$ are the labels of start and end positions of the i-th aspect term, q is the number of aspect terms in the input.

2.3.2 Extract Opinion Terms. To extract opinion terms, we first feed the sequence `[CLS] Sentence [SEP] Aspect Label [SEP]` into BERT, and meanwhile feed the sentence into SR-GAT. In the training process, we directly use the label of the aspect term as `Aspect Label`. In the inference process, we use the predicted aspect

[2] We utilize the original dependency tree to model graph when aspects are not labeled.

Table 3: Statistics of the three datasets. #s denotes the number of sentences. #a and #o denote the number of aspect terms and opinion terms. #p means the number of (aspect, opinion) pairs.

Dataset		14res				14lap				15res				16res			
		#s	#a	#o	#p	#s	#a	#o	#p	#s	#a	#o	#p	#s	#a	#o	#p
$\mathcal{D}_{17}$ [36]	train	3044	3699	3484	–	3048	2373	2504	–	1315	1199	1210	–	–	–	–	–
	test	800	1134	1008	–	800	654	674	–	685	542	510	–	–	–	–	–
$\mathcal{D}_{19}$ [8]	train	1627	2643	–	–	1158	1634	–	–	754	1076	–	–	1079	1512	–	–
	test	500	865	–	–	343	482	–	–	325	436	–	–	329	457	–	–
$\mathcal{D}_{20a}$ [24]	train	1300	–	–	2145	920	–	–	1265	593	-	-	923	842	–	–	1289
	test	496	–	–	862	339	–	–	490	318	–	–	455	320	–	–	465
$\mathcal{D}_{20b}$ [44]	train	1266	–	–	2338	906	–	–	1460	605	–	–	1013	857	–	–	1394
	test	492	–	–	994	328	–	–	543	322	–	–	485	326	–	–	514

terms as `Aspect Label`. Then we obtain the final representation $\tilde{h}_o^K$ from BERT that was enhanced by sentiment and syntax information. We feed $\tilde{h}_o^K$ into a non-linear transformation layer, and acquire the start position score $g_{o,s}^i$ and the end position score $g_{o,e}^i$ of the i-th opinion term, respectively. The objective function for extracting opinion terms is formulated as:

$$\mathcal{L}_o = -\sum_{i=1}^{z} y_{o,s}^i log g_{o,s}^i - \sum_{i=1}^{z} y_{o,e}^i log g_{o,e}^i \quad (10)$$

where $y_{o,s}^i$ and $y_{o,e}^i$ are the labels of start and end positions of the i-th opinion term, z is the number of opinion terms in the input.

2.3.3 Predict Sentiment Polarity. To predict sentiment polarity of specific aspect term, we first format the input sequence as `[CLS] Sentence [SEP] Aspect Label [SEP]`. Then we feed the sentence and the dependency tree into SR-GAT and obtain the output representations $\tilde{h}_s^K$ from BERT. Since `[CLS]` token contains the semantic interaction information of the input, we take `[CLS]` representation $\tilde{h}_{s,0}^K$ for predicting the sentiment polarity. Specifically, the output probability of aspect-based sentiment polarity g_s is computed by a non-linear transformation on $\tilde{h}_{s,0}^K$. Finally, the objective function for the sentiment prediction is:

$$\mathcal{L}_s = -(y_s log(g_s) - (1 - y_s) log(1 - g_s)) \quad (11)$$

2.3.4 Learning Objective. We jointly optimize aspect terms extraction, opinion terms extraction, and sentiment polarity prediction components. The final objective function is formulated as:

$$\mathcal{L}_{ft} = \mathcal{L}_a + \mathcal{L}_o + \mathcal{L}_s \quad (12)$$

2.4 Model Inference

After the above multi-task fine-tuning process, we can test our unified model on all ABSA tasks including AE, OE, ASC, AESC, AOE, AOPE, and ASTE.

It is worth nothing that our fine-tuning process can be performed in an end-to-end manner since the ground truth of all aspect labels is available. During the inference process, we utilize the original dependency tree to model the graph for AE and OE tasks because all aspect labels are unknown. For OE task, we construct the input sequence of BERT as {`[CLS] Sentence [SEP] Opinion [SEP]`}, instead of {`[CLS] Sentence [SEP] Aspect Label [SEP]`}. For ASC task, we directly use the given aspect label to construct the input sequence `[CLS] Sentence [SEP] Aspect Label [SEP]` for BERT. For other tasks, we first predict the aspect label with BERT, then feed the sequence `[CLS] Sentence [SEP] Aspect Label [SEP]` into BERT where `Aspect Label` is the predicted label.

3 EXPERIMENTS

Datasets and Evaluation Metrics. We evaluate our model on three popular benchmarks from the Semeval Challenges [26, 27]. The first dataset only contains the aspect labels which are annotated by Wang et al. [36]. The second dataset is annotated with aspect terms and their corresponding opinion terms by Fan et al. [8]. The third dataset is from Peng et al. [24] that contains aspect labels, their corresponding opinion labels, and sentiment polarities of specific aspect terms. We further introduce the refined version of the third dataset proposed by Xu et al. [44] which removes the triples with inaccurate sentiments and labels the missing triples. These datasets are about `Laptop` and `Restaurant` reviews. To distinguish datasets above, we name these benchmarks according to their published years, i.e. $\mathcal{D}_{17}$, $\mathcal{D}_{19}$, $\mathcal{D}_{20a}$ and $\mathcal{D}_{20b}$. We present the statistics of the three datasets in Table 3. Following previous studies [8, 24], we use metrics to the corresponding ABSA tasks and datasets. These metrics include the precision (P), recall (R), and macro-f1 (F1) scores.

Implementation Details. We run all experiments on the NVIDIA Tesla-V100. Following previous works [21], we utilize English uncased $\text{BERT}_{\texttt{base}}$ (110M) as the backbone for AE, SC, and ASC tasks, and apply English uncased $\text{BERT}_{\texttt{large}}$ (350M) for other four tasks. We select Biaffine Parser [7] for dependency parsing in our model. In the SR-GAT, the dimension of the dependency relation embedding d_p is set to 300. The dimension of POS embedding is set to 300. We vary the layer of SR-GAT (K) in $\{1, 2, 3, 4\}$, and find that $K = 3$ is the best choice. We train our model using Adam optimizer [15]. The initial learning rate is $2e^{-5}$ and keeps decaying during training. We set dropout as 0.1 and batch size as 16.

3.1 Competitive Baselines

To thoroughly and clearly evaluate our framework, we separate all competitive baselines into three groups according to the datasets.

The baselines in the first class are conducted on $\mathcal{D}_{17}$ [36], which are proposed for AE, OE, and ASC tasks. **SPAN-BERT** [12] is a span-based extract-then-classify framework, where BERT is the backbone network. **IMN-BERT** [10] is an interactive multi-task learning network that jointly learns aspect terms and aspect-level sentiment polarities. **RACL-BERT** [4] is a relation-aware collaborative learning framework, which applies relation propagation techniques to analyze fine-grained sentiments. **SKEP** [31] conducts

Table 4: Comparison F1 scores for AE, OE, and ASC tasks on the $\mathcal{D}_{17}$ dataset. The baseline results are directly taken from Mao et al. [21] and Yan et al. [45]. ‡ means $\text{BERT}_{\texttt{large}}$ as backbone for a fair comparison. We highlight the best results in bold.

	14res			14lap			15res		
	AE-F1	OE-F1	ASC-F1	AE-F1	OE-F1	ASC-F1	AE-F1	OE-F1	ASC-F1
SPAN-BERT	86.71	–	71.75	82.34	–	62.50	74.63	–	50.28
IMN-BERT	84.06	85.10	75.67	77.55	81.00	75.56	69.90	73.29	70.10
RACL-BERT	86.38	87.18	81.61	81.79	79.72	73.91	73.99	76.00	74.91
Dual-MRC	86.60	–	82.04	82.51	–	75.97	75.08	–	73.59
Gene-Unified	87.07	87.29	75.56	83.52	77.86	76.76	75.48	76.49	73.91
SK2	**88.82**	**89.91**	**87.64**	**86.17**	**81.04**	**81.66**	**79.75**	**79.43**	**80.72**
SKEP/SK2‡	–	–	91.09/**91.60**	–	–	82.57/**83.32**	–	–	**83.00**/82.19

Table 5: Comparison F1 scores for AOE task on the $\mathcal{D}_{19}$ dataset. Baseline results are directly taken from Mao et al. [21] and Yan et al. [45]. Numbers in bold indicate the best results.

	14res			14lap			15res			16res		
	P	R	F1	P	R	F1	P	R	F1	P	R	F1
IOG	82.83	78.25	80.23	73.43	68.74	70.99	72.19	71.76	71.91	84.36	79.08	81.60
LOTN	84.00	80.52	82.21	77.08	67.62	72.02	76.61	70.29	73.29	86.57	80.89	83.62
Dual-MRC	**89.79**	78.43	83.73	78.21	81.66	79.90	77.19	71.98	74.50	86.07	80.77	83.33
Gene-Unified	86.01	84.76	85.38	83.11	78.13	80.55	**80.12**	80.93	80.52	**89.22**	86.67	87.92
SK2	87.33	**87.93**	**88.54**	**83.32**	**82.63**	**84.97**	75.42	**81.68**	**83.34**	87.17	**87.75**	**88.46**

sentiment masking and three sentiment knowledge prediction objectives to pre-training for ASC task[3]. **Dual-MRC** [21] is a joint training method for a series of ABSA tasks, which converts ABSA tasks into two machine reading comprehension problems. **Unified-Gene** [45] exploits the pre-training sequence-to-sequence model to solve all ABSA subtasks in an end-to-end framework.

The second group of baselines is implemented in $\mathcal{D}_{19}$ [8], which is proposed for the AOE task. **IOG** [8] constructs an encoder to learn aspect-oriented sentence representations and pass them to the decoder for labeling opinion terms, which proposes the AOE task. **LOTN** [40] incorporates position embedding of aspect terms, then uses a BiLSTM to extract the opinion terms.

The baselines in the third group are implemented on $\mathcal{D}_{20a}$ [24] and $\mathcal{D}_{20b}$ [44] for AESC, AOPE, and ASTE tasks. **CMLA** [36] structures coupled attentions to exploit the correlations among sequence tokens for extracting aspect and opinion terms. **Li-unified-R** [24] is a modified model variant of Li-unified [16]. **Peng-two-stage** [24] is a two-stage framework for ASTE task. The first stage predicts aspect terms, opinion terms, and sentiment polarities, and the second stage pairs up the predicted aspect and opinion terms. **PASTE** [22] designs a tagging-free method based on the pre-trained BERT with the review corpus. **Joint-ABSA** [14] constructs a dual-encoder framework that can capture the difference between the subtasks. **Span-ASTE** [43] designs a dual-channel span pruning strategy by incorporating supervision from the aspect term extraction and opinion term extraction tasks.

3.2 Main Evaluations

Evaluations on Three Fundamental Tasks. Table 4 reports the evaluation results of AE, OE, and ASC tasks on $\mathcal{D}_{17}$ [36]. From the results, we can observe that 1) our framework gains significant improvement over all baselines. Specifically, on AE, OE, and ASC tasks, our method achieves 2.89%, 1.87%, and 5.43% average absolute improvements on F1 score, respectively; 2) ASC, as a more useful task in ABSA, gets the most significant improvement. This can be attributed to the general sentiment and aspect-oriented syntax knowledge acquired in our model to facilitate precise interactions between aspect and opinion terms for polarities. 3) SK2 outperforms SKEP model in most scenarios. This demonstrates that the superiority of our introduced knowledge, that is, though utilizing the same sentiment masking with SKEP, our model also considers the conjunctive words and attribute nouns in the post-training procedure and syntax information in the fine-tuning phase.

Evaluations on Aspect & Opinion Extraction Tasks. To further prove the effectiveness of our framework, we also conduct experiments on the AOE task in Table 5. It is shown that our method acquires 2.73% average improvements on $\mathcal{D}_{19}$ [8], which demonstrates the effectiveness of our proposed SK2 on the AOE task. Coupled with the opinion extraction task (i.e., OE task) in Table 4, it is verified that our integrated knowledge can improve the model to capture the interrelatedness between aspect and opinion terms and thus boost the extraction performance.

Evaluations on More Compound Tasks. We show the results of AESC, AOPE, and ASTE tasks on $\mathcal{D}_{20a}$ [24] and $\mathcal{D}_{20b}$ [44] in Table 6. It demonstrates that: First, our span-based approach performs much better than the unified tagging schema, e.g., Li-unified-R and Peng-two-stage, since determining the start and end positions is easier than labeling each token in the input. Second, on extracting (aspect term, opinion term) task, i.e., the AOPE task, our method achieves significant improvements compared with all baselines, which proves the power of our incorporated explicit syntax knowledge for pairing the aspect terms and opinion terms. Third, in the hardest ASTE task, SK2 achieves 2.41% and 1.53% average absolute improvements on the F1 at $\mathcal{D}_{20a}$ and $\mathcal{D}_{20b}$ datasets respectively,

[3]Since we can only acquire the pre-trained SKEP based on $\text{Ernie}_{\texttt{large}}$, we also evaluate SK2 on $\text{BERT}_{\texttt{large}}$ at the ASC task for a fair comparison.

Table 6: Comparison F1 scores for AESC, AOPE, and ASTE tasks on the $\mathcal{D}_{20a}$ and $\mathcal{D}_{20b}$ datasets. Baseline results are directly taken from published papers [14, 21, 22, 43, 45]. † denotes the results are based on $\mathcal{D}_{20b}$ dataset and other results are from $\mathcal{D}_{20a}$ dataset. Numbers in bold indicate the best results.

		14res			14lap			15res			16res		
		P	R	F1	P	R	F1	P	R	F1	P	R	F1
AESC	CMLA	67.80	73.69	70.62	54.70	59.20	56.90	49.90	58.00	53.60	58.90	63.60	61.20
	Li-unified-R	73.15	74.44	73.79	66.28	60.71	63.38	64.95	64.95	64.95	66.33	74.55	70.20
	Peng-two-stage	74.41	73.97	74.19	63.15	61.55	62.34	67.65	64.02	65.79	71.18	72.30	71.73
	Dual-MRC	**76.84**	76.31	76.57	67.45	61.96	64.59	66.84	63.52	65.14	69.18	72.59	70.84
	Gene-Unified	–	–	78.47	–	–	68.17	–	–	69.95	–	–	75.69
	SK2	76.24	**81.72**	**78.72**	**67.91**	**70.84**	**69.42**	**70.84**	**71.93**	**73.30**	**76.61**	**78.98**	**77.78**
AOPE	CMLA	45.17	53.42	48.95	42.10	46.30	44.10	42.70	46.70	44.60	52.50	47.90	50.00
	Li-unified-R	44.37	73.67	55.34	52.29	52.94	52.56	52.75	61.75	56.85	46.11	64.55	53.75
	Peng-two-stage	47.76	68.10	56.10	50.00	58.47	53.85	49.22	65.70	56.23	52.35	70.50	60.04
	Dual-MRC	76.23	73.67	74.93	65.43	61.43	63.37	**72.43**	58.90	64.97	**77.06**	74.41	75.71
	Gene-Unified	–	–	77.68	–	–	66.11	–	–	67.98	–	–	77.38
	SK2	**76.57**	**79.88**	**78.19**	**67.02**	**65.24**	**68.12**	69.28	**73.29**	**72.05**	76.94	**79.07**	**79.89**
ASTE	CMLA	40.11	46.63	43.12	31.40	34.60	32.90	34.40	37.60	35.90	43.60	39.80	41.60
	Li-unified-R	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-two-stage	44.18	62.99	51.89	40.40	47.24	43.50	40.97	54.68	46.79	46.76	62.97	53.62
	Dual-MRC	**71.55**	69.14	70.32	57.39	53.88	55.58	**63.78**	51.87	57.21	68.60	66.24	67.40
	Gene-Unified	–	–	72.46	–	–	57.59	–	–	60.11	–	–	69.98
	SK2	71.40	**74.15**	**73.32**	**57.70**	**58.84**	**60.14**	61.14	**55.45**	**64.32**	**68.85**	**68.68**	**72.03**
	PASTE†	68.70	63.80	66.10	59.70	55.30	57.40	**63.60**	59.80	61.60	68.00	67.70	67.80
	Joint-ABSA†	67.95	71.23	69.55	62.12	56.38	59.11	58.55	60.00	59.27	70.65	70.23	70.44
	Span-ASTE†	**72.89**	70.89	71.85	**63.44**	55.84	59.38	62.18	64.45	63.27	69.45	71.17	70.26
	SK2†	71.48	**75.15**	**73.27**	59.12	**62.06**	**60.56**	62.93	**67.22**	**65.00**	**70.74**	**74.85**	**72.19**

Table 7: Ablation study on ASTE task at the $\mathcal{D}_{20a}$ dataset.

	14res	14lap	15res	16res
SK2	73.32	60.14	64.32	72.03
w/o CWP	71.76	57.47	61.81	71.45
w/o SPP	71.18	57.07	61.60	69.92
w/o ANP-SMLM	71.84	58.21	61.44	70.19
w/o SR-GAT	73.06	59.75	63.66	71.49

demonstrating the effectiveness of implicit sentiment knowledge and explicit syntax knowledge in our framework. Meanwhile, our SK2 outperforms PASTE that is pre-trained with MLM and NSP tasks, which proves the superiority of our introduced knowledge. Thereby, superior performance on all these complicated compound tasks verifies the strong power of our model on ABSA tasks.

In summary, our SK2 is competent in all 7 ABSA tasks on benchmarks and outperforms its competitive baselines. This is achieved by our general and advanced knowledge integration techniques.

3.3 Model Analysis

Ablation Study. To investigate the impact of incorporated implicit sentiment knowledge and explicit syntax knowledge, we conduct a comprehensive ablation study on the ASTE task in Table 7. To verify the effectiveness of our implicit sentiment knowledge, we remove each self-supervised task individually from our framework and denote the framework as "w/o Γ", where Γ includes CWP, SPP, ANP, and SMLM tasks. It is observed that each self-supervised task is useful as removing each of them causes notable performance drop. Particularly, the ANP and SPP tasks play important roles in improving ABSA tasks. The reason might be that the ANP task can help the model to understand aspect terms and the SPP task can provide weak supervision for our framework to predict sentiment polarities. To demonstrate the power of explicit syntax knowledge, we further remove the SR-GAT. Experimental results verify the effectiveness of the explicit aspect-oriented syntax knowledge.

Table 8: Comparison of different trees.

		14res	14lap	15res	16res
AESC	Tree_O	76.26	65.98	72.09	73.02
	Tree_R	78.33	68.43	70.34	76.23
	Tree_{SR}	**78.72**	**69.42**	**73.30**	**77.78**
AOPE	Tree_O	75.34	66.61	69.28	75.40
	Tree_R	77.44	**68.54**	71.24	76.14
	Tree_{SR}	**78.19**	68.12	**72.05**	**79.89**
ASTE	Tree_O	70.98	58.26	62.63	66.89
	Tree_R	**73.54**	57.48	63.56	69.52
	Tree_{SR}	73.32	**60.14**	**64.32**	**72.03**

Comparison of different trees. We compare our spare aspect-oriented dependency tree (Tree_{SR}) with other dependency trees, namely Tree_O and Tree_R. The Tree_O is the original dependency tree of input and Tree_R is the aspect-oriented dependency tree from Wang et al. [34]. As we can see in Table 8, after replacing our tree by Tree_O or Tree_R, the performance of SK2 drops, which demonstrates the superiority of our spare aspect-oriented dependency tree. The main reason may be that the focus of ABSA tasks is the aspect terms instead of the root of dependency tree and some unessential dependency relationships in Tree_O and Tree_R confuse model to extract triples .

Effects of the number of SR-GAT layers. We further study how the number of SR-GAT layers influences the performance of our framework. Figure 4 shows how the performance of our framework changes with respect to the different number of SR-GAT layers on

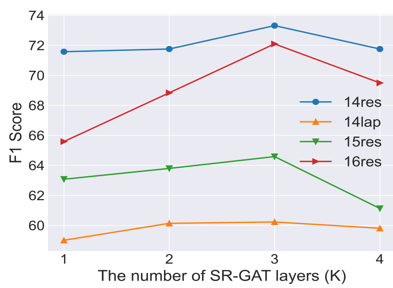


Figure 4: Effects of the number of SR-GAT layers on $\mathcal{D}_{20a}$.

$\mathcal{D}_{20a}$ dataset [24] at ASTE task. We can observe that the performance first increases monotonically until the number of SR-GAT layers reaches 3, and then drops as the number of SR-GAT layers increases. One potential reason is that our aspect-oriented syntax tree has already captured the aspect-opinion in short paths over the tree and three layers of GAT are adequate to model the information.

4 RELATED WORKS

This section groups ABSA tasks into three classes according to their targets, i.e., terms extraction tasks, sentiment prediction tasks, and compound tasks. In each class, we will also focus on the task-specific knowledge integration methods for boosting their performance.

Terms Extraction Tasks. (1) AE focuses on extracting all existing aspect terms from a sentence, which is first proposed by Hu and Liu [11]. Some existing studies treat AE as a sequence tagging task [17, 35, 41], and focus on introducing extra knowledge for better aspect extraction. Xu et al. [41] incorporate domain embeddings and general embeddings into a simple CNN model for acquiring better representation. Li and Lam [17] design an aspect memory and an opinion memory as extra knowledge to extract aspect terms. However, these works neglect to incorporate the knowledge of attribute nouns that reviewers have expressed opinions on. (2) OE aims to extract all opinion terms from the inputs. Wang et al. [36] propose a multilayer attention network to extract opinion terms where each layer consists of a couple of attentions with tensor operators. Li [16] utilize two stacked recurrent neural networks to identify the opinion terms. In most works, the OE task occurs simultaneously with the AE task or as an auxiliary task to improve other ABSA tasks. (3) AOE extracts the opinion terms based on specific aspect terms, which is proposed by Fan et al. [8]. The main challenge of the AOE task is to establish the relations between aspect terms and opinion terms. Wu et al. [39] design a grid tagging scheme to capture the relations, and Ying et al. [47] design two tailor-made opinion transmission mechanisms to grasp opinion term clues. Compared to our framework, these researches usually fail to integrate syntax knowledge for acquiring aspect-related opinion terms. (4) AOPE devotes to extracting the aspect terms and their opinion terms as pairs proposed by Zhao et al. [48]. Some works introduce syntax knowledge for extracting and pairing terms. Wu et al. [38] propose an edge-enhanced syntactic graph convolutional network for enhancing the extraction and pairing of aspect and opinion terms. Wu et al. [37] build a syntax-fusion encoder to encode syntax features for improving pair-wise aspect and opinion terms extraction. But, considering a sentence's complexity, these syntax-enhanced methods do not distinguish important and negligible syntax, which might confuse models to extract and pair terms.

Sentiment Prediction Tasks. ASC predicts aspect-based sentiment polarities in a sentence. Some attention-based works [18, 19] have been explored to calculate the relevancy between the aspect terms and other contextual tokens. Recently, syntax knowledge has been introduced to enhance the ASC task. Chen et al. [3] construct a dependency graph and a latent graph to grasp syntax information. Wang et al. [34] propose an aspect-oriented dependency tree to enhance model to focus on more aspect-related tokens. Despite considering syntax, these works fail to incorporate sentiment knowledge for understanding the polarities of sentiment words.

Compound Tasks. (1) AESC aims to detect the aspect terms and predict their corresponding sentiment polarities in inputs. Traditionally, this task is decoupled into AE and ASC tasks. However, rather than using separate models for each task, most studies address this task by jointly extracting the aspect terms and predicting sentiment polarity in an end-to-end way. Li et al. [16] propose a unified model with maintaining sentiment components. Hu et al. [12] introduce a span-based method, where multiple aspect terms are extracted under the supervision of target span boundaries, and their sentiment polarities are classified using their span representations. Phan and Ogunbona [25] consider the grammatical aspect of the sentence and employ the self-attention mechanism for syntactical learning. Compared with our framework, it can only process the AESC task and is a two-stage method that firstly extracts aspect terms and then predicts their sentiment polarities. (2) ASTE focuses on discovering triples including aspect terms, their sentiment polarities, and opinion terms. Peng et al. [24] is the first to study this task by proposing a two-stage framework. To enhance efficiency, Xu et al. [44] design an end2end model with a position-aware tagging scheme to jointly extract the triples. Xu et al. [43] design a dual-channel span pruning strategy for ASTE task. Mao et al. [21] and Yan et al. [45] construct a unified framework to address all ABSA tasks. Despite the diversity, sentiment-oriented knowledge is seldom exploited but is critical for ABSA tasks.

5 CONCLUSION

This paper considers sentiment and syntax knowledge, which is the key to addressing ABSA tasks. We propose a new unified framework that integrates both implicit sentiment knowledge and explicit syntax knowledge to handle all ABSA tasks. To introduce sentiment knowledge, we devise four self-supervised objectives in terms of sentiment words, attribute nouns, sentiment polarity, and conjunctive words. Besides, we construct an sparse aspect-rooted dependency tree and apply a graph attention network to encode the tree structure, so as to incorporate aspect-oriented syntax knowledge into the framework. We conduct experiments on three public datasets and evaluation results indicate that our proposed framework outputs all compared baselines.

ACKNOWLEDGEMENT

This research is supported by the National Key R&D Program of China under Grant No. 2020AAA0109400, and the National Natural Science Foundation of China under Grant No. 62072007, 62192733, 61832009, 62192731, 62192730.

REFERENCES

[1] Md Shad Akhtar, Asif Ekbal, and Pushpak Bhattacharyya. 2016. Aspect based sentiment analysis: category detection and sentiment classification for Hindi. In *International Conference on Intelligent Text Processing and Computational Linguistics*. Springer, 246–257.

[2] Liang Bin, Liu Quan, Xu Jin, Zhou Qian, and Zhang Peng. 2017. Aspect-based sentiment analysis based on multi-attention CNN. *Journal of Computer Research and Development* 54, 8 (2017), 1724.

[3] Chenhua Chen, Zhiyang Teng, and Yue Zhang. 2020. Inducing Target-Specific Latent Structures for Aspect Sentiment Classification. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. 5596–5607.

[4] Zhuang Chen and Tieyun Qian. 2020. Relation-aware collaborative learning for unified aspect-based sentiment analysis. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. 3685–3694.

[5] Marie-Catherine De Marneffe and Christopher D Manning. 2008. *Stanford typed dependencies manual*. Technical Report. Technical report, Stanford University.

[6] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805* (2018).

[7] Timothy Dozat and Christopher D Manning. 2016. Deep biaffine attention for neural dependency parsing. *arXiv preprint arXiv:1611.01734* (2016).

[8] 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)*. 2509–2518.

[9] Felix Gräßer, Surya Kallumadi, Hagen Malberg, and Sebastian Zaunseder. 2018. Aspect-based sentiment analysis of drug reviews applying cross-domain and cross-data learning. In *Proceedings of the 2018 International Conference on Digital Health*. 121–125.

[10] 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. *arXiv preprint arXiv:1906.06906* (2019).

[11] Minqing Hu and Bing Liu. 2004. Mining and summarizing customer reviews. In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*. 168–177.

[12] Minghao Hu, Yuxing Peng, Zhen Huang, Dongsheng Li, and Yiwei Lv. 2019. Open-domain targeted sentiment analysis via span-based extraction and classification. *arXiv preprint arXiv:1906.03820* (2019).

[13] Sadaf Hussain Janjua, Ghazanfar Farooq Siddiqui, Muddassar Azam Sindhu, and Umer Rashid. 2021. Multi-level aspect based sentiment classification of Twitter data: using hybrid approach in deep learning. *PeerJ Computer Science* 7 (2021), e433.

[14] Hongjiang Jing, Zuchao Li, Hai Zhao, and Shu Jiang. 2021. Seeking Common but Distinguishing Difference, A Joint Aspect-based Sentiment Analysis Model. *arXiv preprint arXiv:2111.09634* (2021).

[15] Diederik P Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *International Conference on Learning Representations*.

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

[17] Xin Li and Wai Lam. 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*. 2886–2892.

[18] Yuncong Li, Cunxiang Yin, Sheng-hua Zhong, and Xu Pan. 2020. Multi-instance multi-label learning networks for aspect-category sentiment analysis. *arXiv preprint arXiv:2010.02656* (2020).

[19] Dehong Ma, Sujian Li, Xiaodong Zhang, and Houfeng Wang. 2017. Interactive attention networks for aspect-level sentiment classification. *arXiv preprint arXiv:1709.00893* (2017).

[20] Vikas Malik and Amit Kumar. 2018. Sentiment Analysis of Twitter Data Using Naive Bayes Algorithm. *International Journal on Recent and Innovation Trends in Computing and Communication* 6, 4 (2018), 120–125.

[21] Yue Mao, Yi Shen, Chao Yu, and Longjun Cai. 2021. A joint training dual-mrc framework for aspect based sentiment analysis. *arXiv preprint arXiv:2101.00816* (2021).

[22] Rajdeep Mukherjee, Tapas Nayak, Yash Butala, Sourangshu Bhattacharya, and Pawan Goyal. 2021. PASTE: A Tagging-Free Decoding Framework Using Pointer Networks for Aspect Sentiment Triplet Extraction. *arXiv preprint arXiv:2110.04794* (2021).

[23] Avinash Chandra Pandey, Dharmveer Singh Rajpoot, and Mukesh Saraswat. 2017. Twitter sentiment analysis using hybrid cuckoo search method. *Information Processing & Management* 53, 4 (2017), 764–779.

[24] 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*, Vol. 34. 8600–8607.

[25] Minh Hieu Phan and Philip O Ogunbona. 2020. Modelling context and syntactical features for aspect-based sentiment analysis. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. 3211–3220.

[26] Maria Pontiki, Dimitrios Galanis, Haris Papageorgiou, Ion Androutsopoulos, Suresh Manandhar, Mohammad Al-Smadi, Mahmoud Al-Ayyoub, Yanyan Zhao, Bing Qin, Orphée De Clercq, et al. 2016. Semeval-2016 task 5: Aspect based sentiment analysis. In *International workshop on semantic evaluation*. 19–30.

[27] Maria Pontiki, Dimitrios Galanis, Harris 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)*. 486–495.

[28] Raksmey Sann and Pei-Chun Lai. 2020. Understanding homophily of service failure within the hotel guest cycle: Applying NLP-aspect-based sentiment analysis to the hospitality industry. *International Journal of Hospitality Management* 91 (2020), 102678.

[29] Chi Sun, Luyao Huang, and Xipeng Qiu. 2019. Utilizing BERT for aspect-based sentiment analysis via constructing auxiliary sentence. *arXiv preprint arXiv:1903.09588* (2019).

[30] Kai Sun, Richong Zhang, Samuel Mensah, Yongyi Mao, and Xudong Liu. 2019. Aspect-level sentiment analysis via convolution over dependency tree. 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)*. 5679–5688.

[31] Hao Tian, Can Gao, Xinyan Xiao, Hao Liu, Bolei He, Hua Wu, Haifeng Wang, and Feng Wu. 2020. SKEP: Sentiment knowledge enhanced pre-training for sentiment analysis. *arXiv preprint arXiv:2005.05635* (2020).

[32] Peter D Turney. 2002. Thumbs up or thumbs down? Semantic orientation applied to unsupervised classification of reviews. *arXiv preprint cs/0212032* (2002).

[33] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems* 30 (2017).

[34] Kai Wang, Weizhou Shen, Yunyi Yang, Xiaojun Quan, and Rui Wang. 2020. Relational graph attention network for aspect-based sentiment analysis. *arXiv preprint arXiv:2004.12362* (2020).

[35] Wenya Wang, Sinno Jialin Pan, Daniel Dahlmeier, and Xiaokui Xiao. 2016. Recursive neural conditional random fields for aspect-based sentiment analysis. *arXiv preprint arXiv:1603.06679* (2016).

[36] 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 AAAI Conference on Artificial Intelligence*, Vol. 31.

[37] Shengqiong Wu, Hao Fei, Yafeng Ren, Donghong Ji, and Jingye Li. 2021. Learn from Syntax: Improving Pair-wise Aspect and Opinion Terms Extractionwith Rich Syntactic Knowledge. *arXiv preprint arXiv:2105.02520* (2021).

[38] Shengqiong Wu, Hao Fei, Yafeng Ren, Bobo Li, Fei Li, and Donghong Ji. 2021. High-Order Pair-Wise Aspect and Opinion Terms Extraction With Edge-Enhanced Syntactic Graph Convolution. *IEEE/ACM Transactions on Audio, Speech, and Language Processing* 29 (2021), 2396–2406.

[39] Zhen Wu, Chengcan Ying, Fei Zhao, Zhifang Fan, Xinyu Dai, and Rui Xia. 2020. Grid tagging scheme for aspect-oriented fine-grained opinion extraction. *arXiv preprint arXiv:2010.04640* (2020).

[40] Zhen Wu, Fei Zhao, Xin-Yu Dai, Shujian Huang, and Jiajun Chen. 2020. Latent opinions transfer network for target-oriented opinion words extraction. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 34. 9298–9305.

[41] Hu Xu, Bing Liu, Lei Shu, and Philip S Yu. 2018. Double embeddings and cnn-based sequence labeling for aspect extraction. *arXiv preprint arXiv:1805.04601* (2018).

[42] Lu Xu, Lidong Bing, Wei Lu, and Fei Huang. 2020. Aspect Sentiment Classification with Aspect-Specific Opinion Spans.. In *EMNLP (1)*. 3561–3567.

[43] Lu Xu, Yew Ken Chia, and Lidong Bing. 2021. Learning Span-Level Interactions for Aspect Sentiment Triplet Extraction. *arXiv preprint arXiv:2107.12214* (2021).

[44] Lu Xu, Hao Li, Wei Lu, and Lidong Bing. 2020. Position-aware tagging for aspect sentiment triplet extraction. *arXiv preprint arXiv:2010.02609* (2020).

[45] Hang Yan, Junqi Dai, Xipeng Qiu, Zheng Zhang, et al. 2021. A unified generative framework for aspect-based sentiment analysis. *arXiv preprint arXiv:2106.04300* (2021).

[46] Da Yin, Tao Meng, and Kai-Wei Chang. 2020. Sentibert: A transferable transformer-based architecture for compositional sentiment semantics. *arXiv preprint arXiv:2005.04114* (2020).

[47] Chengcan Ying, Zhen Wu, Xinyu Dai, Shujian Huang, and Jiajun Chen. 2020. Opinion transmission network for jointly improving aspect-oriented opinion words extraction and sentiment classification. In *CCF International Conference on Natural Language Processing and Chinese Computing*. Springer, 629–640.

[48] He Zhao, Longtao Huang, Rong Zhang, Quan Lu, and Hui Xue. 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*. 3239–3248.