End-to-end Aspect-based Sentiment Analysis with Combinatory Categorial Grammar

Yuanhe Tian♠♥, Weidong Chen♠, Bo Hu♠, Yan Song♠†, Fei Xia♥

♠University of Science and Technology of China ♥University of Washington

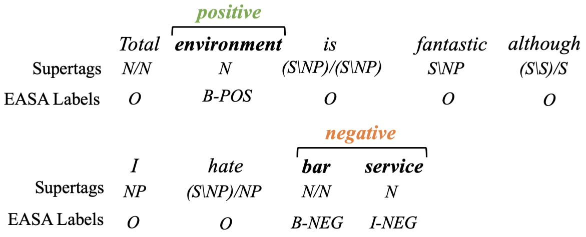
♥{yhtian, fxia}@uw.edu ♠{chenweidong, hubo}@ustc.edu.cn ♠clksong@gmail.com

Abstract

End-to-end Aspect-based Sentiment Analysis (EASA) is a natural language processing (NLP) task that involves extracting aspect terms and identifying the sentiments for them, which provides a fine-grained level of text analysis and thus requires a deep understanding of the running text. Many previous studies leverage advanced text encoders to extract context information and use syntactic information, e.g., the dependency structure of the input sentence, to improve the model performance. However, such models may reach a bottleneck since the dependency structure is not designed to provide semantic information of the text, which is also important for identifying the sentiment and thus leave room for further improvement. Considering that combinatory categorial grammar (CCG) is a formalism that expresses both syntactic and semantic information of a sentence, it has the potential to be beneficial to EASA. In this paper, we propose a novel approach to improve EASA with CCG supertags, which carry the syntactic and semantic information of the associated words and serve as the most important part of the CCG derivation. Specifically, our approach proposes a CCG supertag decoding process to learn the syntactic and semantic information carried by CCG supertags and use the information to guide the attention over the input words so as to identify important contextual information for EASA. Furthermore, a gate mechanism is used in incorporating the weighted contextual information into the backbone EASA decoding process. We evaluate our approach on three publicly available English datasets for EASA, and show that it outperforms strong baselines and achieves state-of-the-art results on all datasets.[1]

1 Introduction

End-to-end aspect-based sentiment analysis (EASA) is an important task that provides the understanding of attitudes and emotions of individuals towards specific topics or entities on a fine-grained level. In general, EASA is normally required to identify aspect terms in the running text and then predict their sentiment polarities. Therefore, understanding sentence structure is of great importance in locating different parts of a sentence and thus finding the aspect and associated sentiment terms for EASA. For example, in Figure 1, a sentence "*Total environment is fantastic although I hate bar service*" contains two aspect terms, namely, "*environment*" and "*bar service*" and the sentiment polarities towards them are positive and negative, respectively.



	Total	environment	is	fantastic	although
Supertags	N/N	N	(S\NP)/(S\NP)	S\NP	(S\S)/S
EASA Labels	O	B-POS	O	O	O

	I	hate	bar	service
Supertags	NP	(S\NP)/NP	N/N	N
EASA Labels	O	O	B-NEG	I-NEG

Figure 1: An example sentence for EASA where the sentiments of two aspect terms, namely, "*environments*" and "*bar service*", are *positive* and *negative*, respectively. The supertag and EASA label of each word are also presented for better illustration. The EASA labels aggregate the BIO labels for aspect term extraction with the positive/negative labels for sentiment polarity.

Most of the previous studies (Hu et al., 2019; He et al., 2019; Luo et al., 2019; Wang et al., 2021; Bie and Yang, 2021; Li et al., 2019a,b; Hu et al., 2019; Chen et al., 2020; Liang et al., 2021) on EASA are generally categorized into three categories, namely, pipeline, multi-task, and joint-label approaches, based on how they formalize the task. Among the three types of approaches, the joint-label approaches aggregate labels for the sub-tasks rather than directly conducting them, and they achieve the best performance. Figure 1 shows the aggregated EASA labels where the first part indicates the position of a word in an aspect term following the

†Corresponding author.

[1] The code involved in this paper is released at https://github.com/synlp/ASA-CCG.

Findings of the Association for Computational Linguistics: ACL 2023, pages 13597–13609
July 9-14, 2023 ©2023 Association for Computational Linguistics

BIO schema and the second part indicates the sentiment of the aspect term.[2] In performing EASA, previous studies try advanced encoders, such as LSTM, Transformer (Vaswani et al., 2017), and BERT (Devlin et al., 2019), to capture contextual information and achieve outstanding performance. Since most aspect terms are noun phrases and the syntactic structure of the text is able to provide additional clues about the sentiment expressed towards an aspect, previous studies leverage syntactic information, e.g., the dependency of the sentence, to further enhance sentiment analysis (Huang and Carley, 2019; Tian et al., 2021a,b; Wu et al., 2021; Liang et al., 2021). However, approaches enhanced by conventional syntactic information, especially the dependencies, may reach a bottleneck since it is unable to provide semantic information about the sentence, which is also important for EASA.

Combinatory categorial grammar (CCG) offers an alternative to phrase structure grammar for describing the syntax and build transparent connections for syntax and semantics (Steedman, 1987; Baldridge, 2002; Hockenmaier and Steedman, 2005). In using combinatory rules over simple syntactic categories, a key feature of CCG is its use of type-logical semantics, which provides a systematic way of associating meanings with the syntactic structures generated by the grammar. This allows for a more precise and intuitive representation of the meanings of sentences as well as each word in it. Particularly, the lexical category (which is also known as CCG supertags) associated with each word conveys both syntactic and semantic information of the word. Therefore, learning the CCG supertagging process allows the model to learn the syntactic and semantic function of each word in the running text and thus shows its effectiveness in many tasks (Lewis et al., 2015; Kasai et al., 2019; Tian and Song, 2022).

In this study, we hypothesize that CCG could be useful in enhancing the performance of the model for EASA as well. We propose a joint-label approach following the encoding-decoding paradigm to enhance EASA with CCG supertags. In doing so, we enhance EASA with a CCG supertag decoding process to learn from the CCG supertags automatically annotated by an off-the-shelf CCG supertagger. An attention mechanism is performed over all input words to identify the ones that contribute to the EASA task, where the attention weights are guided by the supertag decoding process. This allows our model to learn the syntactic and semantic information carried by CCG supertags through the CCG decoding process rather than using them as additional input features and thus makes our model run faster in inference. Furthermore, considering that there could be noise in the auto-generated CCG supertags, we introduce a gate mechanism to balance the contributions between the context information obtained from the text encoder and the attention module. Experimental results on three English benchmark datasets for EASA present the effectiveness of our approach, where our approach outperforms strong baselines and achieves state-of-the-art performance.

2 Related Work

The EASA task has drawn much attention in recent years and existing approaches can be categorized into three groups, namely, pipeline approaches, multi-task approaches, and joint approaches. Specifically, the pipeline approaches (Mitchell et al., 2013; Zhang et al., 2015; Hu et al., 2019) contains two steps, where the first performs the aspect term extraction and the second aims to predict the sentiment polarities of the extracted aspect terms. The multi-task approaches (Ma et al., 2018; Luo et al., 2019; He et al., 2019; Wang et al., 2021) normally use a text encoder to model the input and employ multi-task learning with two separately decoding processes to extract the aspect terms and predict the sentiments. Joint-label approaches (Li and Lu, 2017; Li et al., 2019a; Chen et al., 2020) perform the aspect term extraction and the sentiment polarity prediction tasks simultaneously through a unified labeling scheme.

Most recent approaches to EASA apply advanced encoders (e.g., LSTM, Transformer, and BERT) and achieve promising performance. To further enhance EASA, Chen et al. (2020) proposed a joint-label model to leverage the dependencies between words through graph-based models; Wang et al. (2021) used a hierarchical architecture to perform multi-task learning and outperforms existing multi-task based approaches. Overall, the pipeline approaches often suffer from error propagation issues in the cascade steps and the multi-task approaches have label mismatching problems, where the decoding results usually do not match each other. In comparison, joint-label approaches are

[2] For example, "O" indicates that the associated word is not part of an aspect term and "B-NEG" means the word is the beginning of an aspect term whose sentiment is negative.

superior in avoiding error propagation and label mismatch problems.

Compared with previous studies, this paper proposes a joint-label approach for EASA with the enhancement of using CCG information (rather than the dependencies which are widely used in previous studies) through CCG supertag decoding, attention mechanism, and gate modules.

3 Preliminaries

3.1 Combinatory Categorial Grammar

Combinatory categorial grammar (CCG) is a linguistic framework that describes the syntax and semantics of natural language using mathematical functions. It is based on the idea that the meaning of a sentence can be derived from the combination of its individual words, and each word is assigned a specific category (i.e., supertag) based on its grammatical function. CCG uses a set of rules to combine these categories in order to generate the meaning of a sentence.

The advantage of CCG for NLP is its ability to handle the complex structure of natural language. CCG provides a systematic and rigorous way to analyze the syntactic and semantic structure of sentences, which is essential for building accurate and efficient NLP systems. It also allows for the integration of syntactic and semantic information, which is crucial for language understanding. For example, in the clause "*I hate bar service*" in Figure 1, its CCG derivation presents that "*bar service*" is a nominal phrase that serves as the object and patient of the predicate "*hate*" that convey the negative sentiment, which implies a negative sentiment towards the aspect term "*bar service*".

Comparing with widely used phrase structure grammar (PSG) or dependency grammar, CCG is potentially useful for EASA in two aspects. First, CCG is normally lexicalized[3], where the syntactic categories of words and phrases are determined not only by their syntactic function, but also by their specific meaning and usage. For example, verbs can be divided into intransitive, transitive, and ditransitive verbs depending on the number of arguments they take. If the verb's supertag is "S\NP", it means the verb requires only an NP argument from the left (i.e., the subject) and thus it is an intransitive verb; if its supertag is "(S\NP)/NP", then it first requires an NP from the right (i.e., an object) and then an NP from the left (i.e., a subject); thus, it is a transitive one. Therefore, the supertag of a word specifies not only the word's syntactic category but also other relevant information, such as its subcategorization frame and the types of its arguments, which is of great importance in analyzing semantic relations among words. Second, CCG is able to handle long-distance dependencies in an elegant way through combinatory rules, especially when it uses "*type raising*", a process that changes the syntactic categories of a word or a phrase for representing the complex syntactic structure of a sentence, and thus captures important contextual information for EASA.

3.2 Joint-label Approaches for EASA

Previous joint-label approaches formalize EASA as a standard sequence labeling task, where an input sentence with n words, namely, $\mathcal{X} = x_1 \cdots x_i \cdots x_n$, is tagged with the corresponding EASA joint label sequence $\widehat{\mathcal{Y}} = \widehat{y}_1 \cdots \widehat{y}_i \cdots \widehat{y}_n$ with $\widehat{y}_i$ denotes the joint label for x_i. Herein, a joint label consists of two parts: the first part refers to the BIO label with respect to aspect term boundary and the second part indicates the sentiment polarity (i.e., positive, negative, neutral) of that aspect.

4 The Approach

Our approach for EASA follows the standard sequence labeling paradigm, whose architecture is illustrated in Figure 2. Specifically, our approach consists of four parts: (1) the backbone model to predict EASA labels, (2) the supertag decoding process to learn syntactic and semantic information carried by supertags, (3) the attention module to weigh different contextual information, and (4) the gate mechanism to balance the contribution of the backbone model and the attention module to the EASA task. The process of our approach (which is denoted as f) for EASA can be formalized as

$$\widehat{\mathcal{Y}}, \widehat{\mathcal{Y}^S} = f\left(\mathcal{G}\left(\mathcal{A}\left(\mathcal{X}\right), \mathcal{X}\right), \mathcal{S}\left(\mathcal{X}\right)\right) \qquad (1)$$

where $\widehat{\mathcal{Y}^S} = \widehat{y}^s_1 \cdots \widehat{y}^s_i \cdots \widehat{y}^s_n$ stands for the supertag sequence obtained from the supertag decoding process (denoted as $\mathcal{S}$), $\mathcal{G}$ is the gate module, and $\mathcal{A}$ refers to the attention module. In training, the predicted EASA joint labels $\widehat{\mathcal{Y}}$ are compared with the gold standard $\mathcal{Y}^*$ to obtained the EASA loss L_e and the predicted supertags $\widehat{\mathcal{Y}^S}$ are compared with the silver standard $\mathcal{Y}^*$ obtained from a CCG supertagger to compute the supertag loss L_s. The total loss

[3] We refer to lexicalized CCG in this paper, which is the widely used version of CCG for real parsing and applications.

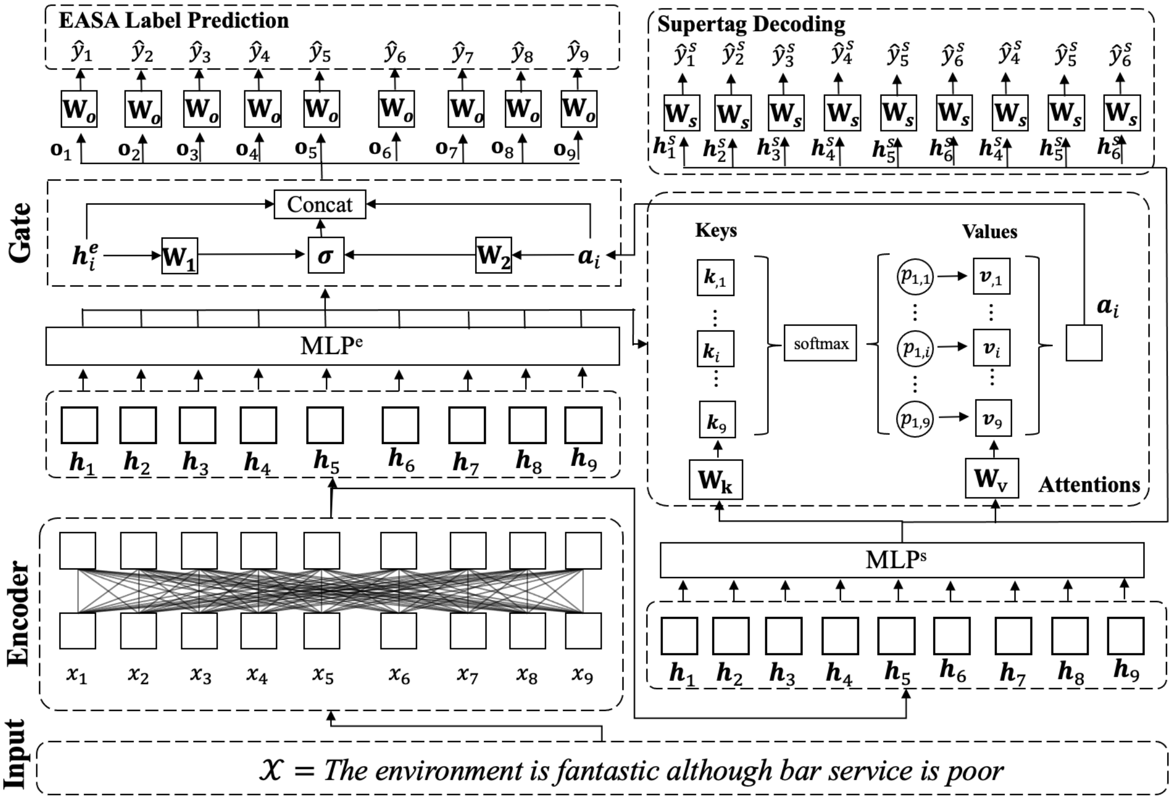


Figure 2: The overall architecture of our model. The right-hand side illustrates the supertag decoding process that learns contextual information by predicting CCG supertags and the attention mechanism that obtains important contextual information guided by the CCG supertags. The left-hand side presents the backbone model for EASA and the gate module to incorporate the contextual information from the attention module into the backbone model.

is the sum of EASA loss and supertag loss and the model is updated accordingly. In this section, we will first introduce the overall tagging process for EASA labels, next present the supertag decoding process, then elaborate the attention module, and finally illustrate the gate mechanism.

4.1 Overall Tagging Process

The overall tagging process follows the standard encoding-decoding paradigm. First, a text encoder f_e (such as BiLSTM, Transformer, and BERT) is applied to the input sentence and obtains the hidden vector sequence

$$\mathbf{h}_1 \cdots \mathbf{h}_i \cdots \mathbf{h}_n = f_e\left(\mathcal{X}\right) \tag{2}$$

where $\mathbf{h}_i$ is the hidden vector of x_i which stores contextual information learnt by the encoder. Then, the hidden vector for each word is fed into a multi-layer perceptron (denoted by MLP^e) to obtain the hidden vector $\mathbf{h}_i^e$ for EASA through $\mathbf{h}_i^e = \text{MLP}^e(\mathbf{h}_i)$. Afterwards, $\mathbf{h}_i^e$ and the output from the attention module $\mathbf{a}_i$ are fed into the gate module to obtain the output $\mathbf{o}_i$ through $\mathbf{o}_i = \mathcal{G}(\mathbf{h}_i^e, \mathbf{a}_i)$. Finally, $\mathbf{o}_i$ is fed into a fully connected layer to produce $\mathbf{u}_i = \mathbf{W}_o \cdot \mathbf{o}_i + \mathbf{b}_o$ with $\mathbf{W}_o$ and $\mathbf{b}_o$ serving as its weight matrix and bias vector. Finally, a *softmax* classifier is applied to the resulting vector $\mathbf{u}_i$ to predict the joint label $\hat{y}_i$ for EASA.

4.2 Supertag Decoding

To leverage the information carried by CCG supertags, one straightforward approach is to use an off-the-shelf CCG supertagger to tag the input sentence and then use the supertags as extra word-level features by concatenating them with the input words before sending them to the text encoder. However, such approaches require the CCG supertagging as a pre-processing step in inference, which is not efficient, especially when the data to be processed is relatively large. Considering combining several decoding processes serves as an effective approach to learning from different tasks and does not require the label from different tasks as an extra input, we propose to learn the CCG information through an additional CCG supertag decoding process and then use the CCG information to guide EASA through an attention mechanism over all input words.

For CCG supertag decoding, we take the hidden

Dataset	LPTP			REST			TWTR
	Train	Test	Total	Train	Test	Total	Total
# of Sentences	3,045	800	3,845	3,877	2,158	6,035	2,350
# of Aspects	2,300	634	2,934	4,310	2,288	6,598	3,223
(a) # of Positive	987	339	1,326	2,609	1,524	4,133	698
(b) # of Negative	861	130	991	1,035	501	1,536	271
(c) # of Neutral	452	165	617	666	263	929	2,254

Table 1: The statistics of three benchmark datasets, showing the numbers of sentences and aspect terms, as well as the numbers of aspect terms with positive (POS), negative (NEG), and neutral (NEU) sentiment polarities.

Hyper-parameters	Values
Learning Rate	5e-6, **1e-5**, 3e-5
Warmup Rate	0.1, **0.2**
Dropout Rate	**0.2**
Batch Size	4, 8, **16**

Table 2: The hyper-parameters that were tested in our experiments. The best ones are highlighted in boldface.

vector $\mathbf{h}_i$ of the word x_i obtained from the encoder and pass it through an MLP (denoted by MLP^s) to get $\mathbf{h}_i^s = \text{MLP}^s(\mathbf{h}_i)$. The resulting $\mathbf{h}_i^s$ is then mapped to the vector $\mathbf{u}_i^s = \mathbf{W}_s \cdot \mathbf{h}_i^s + \mathbf{b}_s$ in the output space by a trainable matrix $\mathbf{W}_s$ and a bias vector $\mathbf{b}_s$. Finally, a *softmax* classifier is applied to $\mathbf{u}_i^s$ to predict the supertag $\widehat{y}_i^s$.

4.3 Supertag-driven Attentions

In the attention module, we use two trainable matrix $\mathbf{W}_k$ and $\mathbf{W}_v$ to map $\mathbf{h}_j^s$ to the key vector $\mathbf{k}_j$ and value vector $\mathbf{v}_j$, respectively:

$$\mathbf{k}_j = \mathbf{W}_k \cdot \mathbf{h}_j^s \tag{3}$$

and

$$\mathbf{v}_j = \mathbf{W}_v \cdot \mathbf{h}_j^s \tag{4}$$

Then, for each word x_i, we compute the attention weight $p_{i,j}$ assigned to the value $\mathbf{v}_j$ through

$$p_{i,j} = \frac{\exp\left(\mathbf{h}_i^e \cdot \mathbf{k}_j\right)}{\sum_{l=1}^{n} \exp\left(\mathbf{h}_i^e \cdot \mathbf{k}_l\right)} \tag{5}$$

Afterwards, we apply $p_{i,j}$ to the value vector $\mathbf{v}_j$ and obtain the weighted sum vector $\mathbf{a}_i$ via

$$\mathbf{a}_i = \sum_{j=1}^{n} p_{i,j} \cdot \mathbf{v}_j \tag{6}$$

Finally, $\mathbf{a}_i$ is fed into the gate module.

In training, the model is optimized on EASA and CCG supertagging, which allows our model to learn CCG information and use it to enhance the entity representation through the attention mechanism with the attention weights assigned to different input words guided by the learned CCG information.

4.4 The Gate Mechanism

We observe that the contribution of the obtained contextual information to the EASA task could vary in different contexts and a gate module (denoted by $\mathcal{G}$) is naturally desired to weight such information in varying contexts. Thus, to improve the capability of EASA with the semantic information, we propose a gate module to aggregate such information to the backbone NER model. Particularly, we use a $reset$ gate to control the information flow by

$$\mathbf{g}_i = \sigma(\mathbf{W}_1 \cdot \mathbf{h}_i^e + \mathbf{W}_2 \cdot \mathbf{a}_i + \mathbf{b}), \tag{7}$$

where $\mathbf{W}_1$ and $\mathbf{W}_2$ are trainable matrices and $\mathbf{b}$ the corresponding bias term. Afterwards, we use

$$\mathbf{o}_i = [\mathbf{g}_i \circ \mathbf{h}_i^e] \oplus [(\mathbf{1} - \mathbf{g}_i) \circ \mathbf{a}_i] \tag{8}$$

to balance the information from the backbone model and the attention module, where $\oplus$ denotes vector concatenation operation, $\mathbf{o}_i$ is the derived output of the gate module; $\circ$ represents the element-wise multiplication operation and $\mathbf{1}$ is a 1-vector with its all elements equal to 1.

5 Experimental Settings

5.1 Datasets

In experiments, we follow previous studies for EASA and evaluate models on three English benchmark datasets, including restaurant (REST) dataset from SemEval ABSA challenges (Pontiki et al., 2014, 2015, 2016), laptop (LPTP) dataset from Pontiki et al. (2014), and Twitter (TWTR) dataset from Mitchell et al. (2013).[4] All datasets contain the ground truth labels of the target aspects and

[4] All datasets are public available and our use of them is consistent with their intended use.

Models	**Dev**			**Test**			**Para. #**	**Speed**
	REST	**LPTP**	**TWTR**	**REST**	**LPTP**	**TWTR**		
BERT-Base	73.62$_{\pm 0.11}$	60.54$_{\pm 0.14}$	55.78$_{\pm 0.08}$	73.44$_{\pm 0.13}$	60.72$_{\pm 0.12}$	55.99$_{\pm 0.12}$	110M	35.4
+ Concat	75.80$_{\pm 0.15}$	61.21$_{\pm 0.09}$	57.74$_{\pm 0.10}$	75.69$_{\pm 0.10}$	61.34$_{\pm 0.13}$	57.88$_{\pm 0.14}$	113M	33.6
+ Att + Gate	73.80$_{\pm 0.12}$	60.68$_{\pm 0.10}$	55.95$_{\pm 0.09}$	73.61$_{\pm 0.14}$	60.90$_{\pm 0.08}$	56.16$_{\pm 0.13}$	112M	30.6
+ Ours	**76.81**$_{\pm 0.10}$	**66.20**$_{\pm 0.13}$	**60.59**$_{\pm 0.10}$	**76.72**$_{\pm 0.11}$	**66.64**$_{\pm 0.15}$	**60.82**$_{\pm 0.12}$	112M	30.9
BERT-Large	76.44$_{\pm 0.13}$	65.30$_{\pm 0.10}$	58.42$_{\pm 0.14}$	76.27$_{\pm 0.11}$	65.43$_{\pm 0.12}$	58.66$_{\pm 0.10}$	336M	24.5
+ Concat	76.72$_{\pm 0.15}$	66.01$_{\pm 0.11}$	58.90$_{\pm 0.10}$	76.50$_{\pm 0.15}$	66.20$_{\pm 0.12}$	59.10$_{\pm 0.12}$	339M	23.0
+ Att + Gate	76.62$_{\pm 0.14}$	65.52$_{\pm 0.16}$	58.50$_{\pm 0.10}$	76.33$_{\pm 0.12}$	65.50$_{\pm 0.13}$	58.73$_{\pm 0.09}$	338M	21.5
+ Ours	**78.01**$_{\pm 0.11}$	**68.42**$_{\pm 0.14}$	**62.08**$_{\pm 0.13}$	**77.95**$_{\pm 0.10}$	**68.68**$_{\pm 0.11}$	**62.37**$_{\pm 0.14}$	338M	21.3
XLNet-Base	73.75$_{\pm 0.12}$	60.76$_{\pm 0.10}$	56.01$_{\pm 0.13}$	73.68$_{\pm 0.11}$	61.09$_{\pm 0.14}$	56.36$_{\pm 0.11}$	118M	34.8
+ Concat	76.40$_{\pm 0.13}$	62.03$_{\pm 0.13}$	58.37$_{\pm 0.15}$	76.29$_{\pm 0.12}$	62.28$_{\pm 0.10}$	58.47$_{\pm 0.12}$	121M	32.6
+ Att + Gate	73.87$_{\pm 0.10}$	60.90$_{\pm 0.14}$	56.26$_{\pm 0.11}$	73.80$_{\pm 0.13}$	61.34$_{\pm 0.10}$	56.52$_{\pm 0.14}$	120M	29.1
+ Ours	**77.14**$_{\pm 0.11}$	**66.33**$_{\pm 0.14}$	**60.80**$_{\pm 0.12}$	**77.09**$_{\pm 0.10}$	**66.80**$_{\pm 0.12}$	**60.92**$_{\pm 0.11}$	120M	29.2
XLNet-Large	76.72$_{\pm 0.12}$	65.60$_{\pm 0.10}$	58.78$_{\pm 0.11}$	76.60$_{\pm 0.13}$	65.85$_{\pm 0.14}$	59.04$_{\pm 0.13}$	362M	23.8
+ Concat	77.21$_{\pm 0.10}$	66.69$_{\pm 0.13}$	59.92$_{\pm 0.14}$	77.15$_{\pm 0.09}$	67.02$_{\pm 0.16}$	60.28$_{\pm 0.13}$	365M	22.1
+ Att + Gate	76.87$_{\pm 0.15}$	65.84$_{\pm 0.11}$	60.06$_{\pm 0.15}$	76.83$_{\pm 0.10}$	66.01$_{\pm 0.14}$	59.27$_{\pm 0.12}$	364M	20.7
+ Ours	**78.51**$_{\pm 0.12}$	**68.64**$_{\pm 0.10}$	**62.77**$_{\pm 0.14}$	**78.24**$_{\pm 0.09}$	**68.93**$_{\pm 0.10}$	**62.90**$_{\pm 0.11}$	364M	20.5

Table 3: Experimental results (average F1 scores of five runs and the standard deviation) of different models on three English benchmark datasets. The last two columns show the number of parameters and the inference speed (in terms of the number of sentences processed per second) of the models, respectively. The improvement of our approach over the baselines is significant under all settings with $p < 0.005$.

their sentiment polarities. Following the convention in previous studies for aspect-level sentiment analysis (Li et al., 2019a,b; He et al., 2019; Hu et al., 2019; Qin et al., 2021, 2022), we only consider three sentiment polarities, namely, *positive*, *negative*, and *neutral*, where all cases labeled by a *conflict* sentiment in REST and LPTP dataset are filtered out. For all datasets, we report in Table 1 their numbers of sentences and aspect terms, as well as the numbers of aspect terms with *positive*, *neutral*, and *negative* sentiment polarities. The TWTR dataset does not have a standard train-test split and thus we only report its overall statistics.

5.2 Baselines

To explore the effect of our approach to leverage CCG information, we compare our approach with the following baselines:

Base: This baseline follows the standard encoding-decoding paradigm, which corresponds to the backbone model of our approach.

Concat: This baseline uses the auto-generated CCG supertags as additional input. Specifically, the embeddings of the supertags are concatenated with the hidden vectors of the corresponding words and the resulting vectors are fed into a two-layer Transformer to further encode contextual information before passing through the decoder.

Attention + Gate: This baseline uses the same attention and gate module as our proposed model but does not use the CCG decoding process to learn the CCG supertag information.

5.3 Implementation

To obtain the auto-generated CCG supertags for the baselines and our approach, we use an off-the-shelf CCG supertagger called NeST-CCG[5] (Tian et al., 2020) to supertag the text. Because a high-quality text representation is significantly important for a model to obtain high performance in many tasks (Mikolov et al., 2013; Song et al., 2017; Bojanowski et al., 2017; Song et al., 2018; Song and Shi, 2018; Peters et al., 2018; Diao et al., 2020; Lewis et al., 2020; Song et al., 2021), in the experiments, we choose two commonly used pre-trained language models, namely BERT (Devlin et al., 2019) and XLNet (Yang et al., 2019). We use the base and large versions of the models following the default hyper-parameter settings, i.e., 12 layers of self-attention with 768-dimensional hidden vectors for the base version and 24 layers of self-attention with 1024 dimensional hidden vectors for the large version.[6]

We randomly initialize all trainable parameters in our approach and update them during training. For REST and LPTP which do not have an offi-

[5] We downloaded the publicly available model from https://github.com/synlp/NeST-CCG following its intended use.

[6] We download the publicly available cased version of BERT from https://github.com/google-research/bert and XLNet from https://github.com/zihangdai/xlnet following their intended use.

Models		REST	LPTP	TWTR
Pipeline	Hu et al. (2019)	74.92	68.06	57.69
Multi-task	He et al. (2019)	-	58.37	-
	Luo et al. (2019)	72.78	60.35	51.37
	Wang et al. (2021)	-	63.69	-
	Bie and Yang (2021)	-	55.08	47.89
Joint-label	Li et al. (2019a)	69.80	57.90	48.01
	Li et al. (2019b)	73.24	61.12	-
	Hu et al. (2019)	57.85	48.66	48.11
	Chen et al. (2020)	77.81	68.53	62.26
	Liang et al. (2021)	-	63.04	-
Ours (BERT-Large)		78.02	68.77	62.47
Ours (XLNet-Large)		**78.29**	**69.01**	**62.98**

Table 4: Comparison of our best-performing models with previous studies.

cial development set, we randomly sampled 10% instances from the training set and use them as the development set for hyper-parameter tuning (the hyper-parameters tried in the experiments are elaborated in Table 2). Once the hyper-parameters are tuned, we train our final model on the whole training set and evaluate it on the test set. For TWTR, we follow the convention in previous studies (Mitchell et al., 2013; Zhang et al., 2015; Li et al., 2019a; Luo et al., 2019; Hu et al., 2019) to use ten-fold cross-validation on it, where the hyper-parameter tuning process is the same as the experiments on REST and LPTP. For evaluation, we follow previous studies (Li et al., 2019a,b; Luo et al., 2019; He et al., 2019; Hu et al., 2019) and evaluate all models with F1 scores.

6 Results and Analysis

6.1 Overall Results

We run our models and baselines using the base and large versions of BERT and XLNet. Table 3 shows the average results (F1 scores) of five runs on the development and test set of the three English datasets, where the number of parameters ("Para. #") and the inference speed ("Speed") in terms of the number of sentences processed per second are shown in the last two columns.

There are several observations from Table 3. First, our approach outperforms the "Base" model with base and large versions of BERT and XLNet on all datasets without requiring many additional model parameters, which is promising given that BERT and XLNet baselines have already achieved outstanding performance on the datasets. Second, compared with the "Concat" baselines that leverage CCG supertags by concatenating the supertag embeddings with the hidden vectors, our approach consistently achieves better performance under all settings, which demonstrates the effectiveness of the attention and gate modules, as well as the CCG supertag decoding process, to leverage CCG information to improve EASA. Third, our model significantly outperforms the "Att+Gate" models without the CCG supertag decoding process to learn the CCG information, which confirms the effectiveness of the CCG supertags in providing necessary information to guide the model to distinguish important context information.

Next, we compare our best-performing models using the large version of BERT and XLNet with previous studies for EASA. The results are in Table 4, showing that our approach outperforms all the previous studies and achieves state-of-the-art performance on the three benchmark datasets. Particularly, different from existing approaches that leverage dependencies for EASA, our approach proposes an alternative by incorporating syntactic and semantic information from CCG supertags into the EASA task. Our model integrates a CCG supertag decoding process to automatically learn the syntactic and semantic information contained within CCG supertags. This eliminates the need for using CCG supertags as additional input features, which is computationally expensive in inference.

6.2 Comparison with Dependencies

Since the dependency structure of the input sentence is widely used in previous studies for EASA,

Models	REST	LPTP	TWTR
BERT-base	73.44	60.72	55.99
+ Dep. (GAT)	76.31	66.03	58.00
+ CCG (Ours)	**76.72**	**66.64**	**60.82**
BERT-large	76.27	65.43	58.66
+ Dep. (GAT)	77.13	68.18	61.88
+ CCG (Ours)	**77.95**	**68.68**	**62.37**
XLNet-base	73.68	61.09	56.36
+ Dep. (GAT)	76.63	66.04	60.27
+ CCG (Ours)	**77.09**	**66.80**	**60.92**
XLNet-large	76.50	66.05	59.24
+ Dep. (GAT)	76.97	66.52	59.85
+ CCG (Ours)	**78.24**	**68.93**	**62.90**

Table 5: Comparison of the results (average F1 scores) of models with GAT to encode dependency information and our models to leverage CCG supertags, where the performance of the “Base” model with BERT and XLNet encoder are also reported for reference.

we compare our approach which leverages CCG supertags with previous studies that leverage dependencies. Since graph attention networks (GAT) (Veličković et al., 2017) is a widely used architecture with an attention mechanism to encode dependencies and is demonstrated to be effective in many NLP tasks, we use GAT to encode the dependency of the input sentence obtained from the dependency parser DMPar (Tian et al., 2022). The average F1 scores of GAT and our approach, as well as the “Base” model, with base and large versions of BERT and XLNet are reported in Table 5, where our approach consistently outperforms the GAT model under all settings. This observation shows that in addition to dependencies, CCG could be another effective linguistic information that could be beneficial for EASA.

6.3 Ablation Study

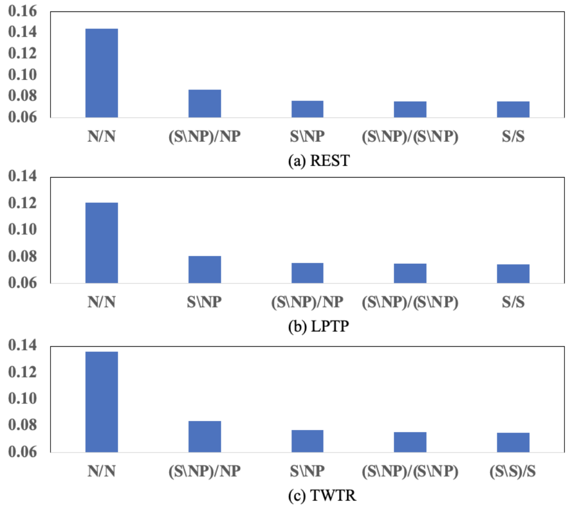
To determine the effect of the attention module and the gate module in leveraging CCG supertag information, we conduct an ablation study where either the attention module or the gate module is ablated from our full model. Specifically, when the attention module is ablated, the attention weights are equal for all the words; when the gate module is ablated, the output from the attention module is directly concatenated with the hidden vector of each word. Table 6 summarizes the average performance of different models, where the performance of the “Base” model with base and large versions of BERT and XLNet is included for reference. The table shows that the ablation of either the attention module or the gate module hurts the model’s performance. Furthermore, the performance decrease due to ablating the attention module is much severe than that due to ablating the gate module, indicating the importance of the attention module in leveraging CCG supertags, as the attentions allow the model to identify the important context features for EASA and leverage them accordingly to improve system performance.

Models	REST	LPTP	TWTR
Full Model	**76.72**	**66.64**	**60.82**
– Att	76.09	63.54	59.03
– Gate	76.53	66.39	60.35
BERT-base	73.44	60.72	55.99
Full Model	**77.95**	**68.68**	**62.37**
– Att	76.78	67.13	60.32
– Gate	77.64	68.37	62.04
BERT-large	76.27	65.43	58.66
Full Model	**77.09**	**66.80**	**60.92**
– Att	76.60	64.74	59.43
– Gate	76.76	66.57	60.61
XLNet-base	73.68	61.09	56.36
Full Model	**78.24**	**68.93**	**62.90**
– Att	77.41	67.53	60.67
– Gate	77.97	68.56	62.59
XLNet-large	76.50	66.05	59.24

Table 6: Experimental results (average F1 scores) of ablation studies, where either the attention module (“– Att”) or the gate module (“– Gate”) is ablated from our full model. The performance of the “Base” model with base and large versions of BERT and XLNet are also reported for reference.

6.4 The Effect of Supertags

To investigate the contribution of individual CCG supertags to the EASA task, for each word that is a part of an aspect term, we collect the attention weights over the input words and their attached CCG supertags and compute the average weight of each supertag. Figure 3 presents the top 5 ranking CCG supertags when using our model with BERT-base encoder on the test set of REST, LPTP, and TWTR. It is observed that, “*N/N*”, “*S\NP*”, “*(S\NP)/NP*”, and “*(S\NP)/(S\NP)*” are the top rank-



(a) REST

(b) LPTP

(c) TWTR

Figure 3: The average attention weight assigned to the top 5 ranking CCG supertags by our model that uses BERT-base encoder on the test set of (a) REST, (b) LPTP, and (c) TWTR.

ing CCG supertags shared by all datasets. One possible explanation could be the following. "*N/N*" is often the supertag of a noun modifier (e.g., the word "*beautiful*" in "*beautiful view*"), which could express sentiment toward the noun being modified. The supertag "*S\NP*" is for both intransitive verbs and adjectives in the predicative position (e.g., "*fantastic*" in "*Total environment is fantastic*" shown in Figure 1); some words in that position could provide sentiment information toward the subject of the clause. The supertag "*(S\NP)/NP*" is normally associated with transitive verbs and some of them (such as *like*, *hate*, and *prefer*) may express sentiment toward the object of the clause. The supertag "*(S\NP)/(S\NP)*" is normally for adverbs that modify VPs. Some of those adverbs (e.g., *happily* and *not*) can be important for identifying the sentiment.

7 Conclusion

In this paper, we propose to enhance joint-label EASA by leveraging CCG information, which contains both syntactic and semantic information of the running text and thus shows its superior to the conventional phrase and dependency style grammars. Specifically, we learn the CCG supertag information through a CCG supertag decoding process, where such information is used to guide the attention weights over the input words in the attention module so that important context information is distinguished and leverage accordingly. To further enhance the model performance, we employ the gate module to balance the contribution of the context information obtained from the backbone text encoder and the attention module. Experimental results and further analysis on three English benchmark datasets demonstrate the effectiveness of the proposed model, where our model outperforms strong baselines and achieves state-of-the-art performance on all datasets. For future study, we plan to explore effective approaches to leverage the CCG derivation of the input text to improve EASA.

Acknowledgements

This work is supported by the 30^{th} research institute of China Electronics Technology Group Corporation Limited under the project "Dynamic Evolution of Target Emotional Characteristics and Behavior Trend Research and Judgment Technology" (30605010103) and Shenzhen Science and Technology Program under the project "Fundamental Algorithms of Natural Language Understanding for Chinese Medical Text Processing" (JCYJ20210324130208022).

References

Jason Baldridge. 2002. Lexically Specified Derivational Control in Combinatory Categorial Grammar.

Yong Bie and Yan Yang. 2021. A Multitask Multiview Neural Network for End-to-end Aspect-based Sentiment Analysis. *Big Data Mining and Analytics*, 4(3):195–207.

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. Enriching Word Vectors with Subword Information. *Transactions of the Association for Computational Linguistics*, 5:135–146.

Guimin Chen, Yuanhe Tian, and Yan Song. 2020. Joint Aspect Extraction and Sentiment Analysis with Directional Graph Convolutional Networks. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 272–279.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep bidirectional Transformers for Language Understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186.

Shizhe Diao, Jiaxin Bai, Yan Song, Tong Zhang, and Yonggang Wang. 2020. ZEN: Pre-training Chinese Text Encoder Enhanced by N-gram Representations. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 4729–4740.

Ruidan He, Wee Sun Lee, Hwee Tou Ng, and Daniel Dahlmeier. 2019. An Interactive Multi-Task Learning Network for End-to-End Aspect-Based Sentiment

Analysis. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 504–515.

Julia Hockenmaier and Mark Steedman. 2005. CCGbank: User's Manual.

Minghao Hu, Yuxing Peng, Zhen Huang, Dongsheng Li, and Yiwei Lv. 2019. Open-Domain Targeted Sentiment Analysis via Span-Based Extraction and Classification. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 537–546.

Binxuan Huang and Kathleen M Carley. 2019. Syntax-Aware Aspect Level Sentiment Classification with Graph Attention Networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 5472–5480.

Jungo Kasai, Dan Friedman, Robert Frank, Dragomir Radev, and Owen Rambow. 2019. Syntax-aware Neural Semantic Role Labeling with Supertags. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 701–709, Minneapolis, Minnesota.

Mike Lewis, Luheng He, and Luke Zettlemoyer. 2015. Joint A* CCG Parsing and Semantic Role Labelling. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1444–1454, Lisbon, Portugal.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: Denoising Sequence-to-Sequence Pretraining for Natural Language Generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online.

Hao Li and Wei Lu. 2017. Learning Latent Sentiment Scopes for Entity-Level Sentiment Analysis. In *Thirty-First AAAI Conference on Artificial Intelligence*.

Xin Li, Lidong Bing, Piji Li, and Wai Lam. 2019a. A Unified Model for Opinion Target Extraction and Target Sentiment Prediction. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 6714–6721.

Xin Li, Lidong Bing, Wenxuan Zhang, and Wai Lam. 2019b. Exploiting BERT for End-to-End Aspect-based Sentiment Analysis. In *Proceedings of the 5th Workshop on Noisy User-generated Text (W-NUT 2019)*, pages 34–41.

Yunlong Liang, Fandong Meng, Jinchao Zhang, Yufeng Chen, Jinan Xu, and Jie Zhou. 2021. A Dependency Syntactic Knowledge Augmented Interactive Architecture for End-to-end Aspect-based Sentiment Analysis. *Neurocomputing*, 454:291–302.

Huaishao Luo, Tianrui Li, Bing Liu, and Junbo Zhang. 2019. DOER: Dual Cross-Shared RNN for Aspect Term-Polarity Co-Extraction. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 591–601.

Dehong Ma, Sujian Li, and Houfeng Wang. 2018. Joint Learning for Targeted Sentiment Analysis. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4737–4742.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient Estimation of Word Representations in Vector Space. *arXiv preprint arXiv:1301.3781*.

Margaret Mitchell, Jacqui Aguilar, Theresa Wilson, and Benjamin Van Durme. 2013. Open Domain Targeted Sentiment. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1643–1654.

Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep Contextualized Word Representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 2227–2237, New Orleans, Louisiana.

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

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

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

Han Qin, Guimin Chen, Yuanhe Tian, and Yan Song. 2021. Improving Federated Learning for Aspect-based Sentiment Analysis via Topic Memories. In

Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, pages 3942–3954, Online and Punta Cana, Dominican Republic.

Han Qin, Yuanhe Tian, Fei Xia, and Yan Song. 2022. Complementary Learning of Aspect Terms for Aspect-based Sentiment Analysis. In *Proceedings of the 13th Language Resources and Evaluation Conference*.

Yan Song, Chia-Jung Lee, and Fei Xia. 2017. Learning Word Representations with Regularization from Prior Knowledge. In *Proceedings of the 21st Conference on Computational Natural Language Learning (CoNLL 2017)*, pages 143–152.

Yan Song and Shuming Shi. 2018. Complementary Learning of Word Embeddings. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI-18*, pages 4368–4374.

Yan Song, Shuming Shi, and Jing Li. 2018. Joint Learning Embeddings for Chinese Words and Their Components via Ladder Structured Networks. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence*, pages 4375–4381.

Yan Song, Tong Zhang, Yonggang Wang, and Kai-Fu Lee. 2021. ZEN 2.0: Continue Training and Adaption for N-gram Enhanced Text Encoders. *arXiv preprint arXiv:2105.01279*.

Mark Steedman. 1987. Combinatory Grammars and Parasitic Gaps. *Natural Language & Linguistic Theory*, 5(3):403–439.

Yuanhe Tian, Guimin Chen, and Yan Song. 2021a. Aspect-based Sentiment Analysis with Type-aware Graph Convolutional Networks and Layer Ensemble. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2910–2922, Online.

Yuanhe Tian, Guimin Chen, and Yan Song. 2021b. Enhancing Aspect-level Sentiment Analysis with Word Dependencies. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 3726–3739, Online.

Yuanhe Tian and Yan Song. 2022. Combinatory Grammar Tells Underlying Relevance among Entities. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 5780–5786, Abu Dhabi, United Arab Emirates.

Yuanhe Tian, Yan Song, and Fei Xia. 2020. Supertagging Combinatory Categorial Grammar with Attentive Graph Convolutional Networks. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6037–6044.

Yuanhe Tian, Yan Song, and Fei Xia. 2022. Enhancing Structure-aware Encoder with Extremely Limited Data for Graph-based Dependency Parsing. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 5438–5449, Gyeongju, Republic of Korea.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is All You Need. In *Advances in neural information processing systems*, pages 5998–6008.

Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. 2017. Graph Attention Networks. *arXiv preprint arXiv:1710.10903*.

Xinyi Wang, Guangluan Xu, Zequn Zhang, Li Jin, and Xian Sun. 2021. End-to-end Aspect-based Sentiment Analysis with Hierarchical Multi-task Learning. *Neurocomputing*, 455:178–188.

Chao Wu, Qingyu Xiong, Hualing Yi, Yang Yu, Qiwu Zhu, Min Gao, and Jie Chen. 2021. Multiple-element Joint Detection for Aspect-Based Sentiment Analysis. *Knowledge-Based Systems*, 223:107073.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. 2019. XLNet: Generalized Autoregressive Pretraining for Language Understanding. In *Advances in Neural Information Processing Systems 32*, pages 5753–5763.

Meishan Zhang, Yue Zhang, and Duy-Tin Vo. 2015. Neural Networks for Open Domain Targeted Sentiment. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 612–621.

ACL 2023 Responsible NLP Checklist

A For every submission:

☑ A1. Did you describe the limitations of your work?
Section 8

☒ A2. Did you discuss any potential risks of your work?
We don't think there is any risk of our work.

☑ A3. Do the abstract and introduction summarize the paper's main claims?
Abstract and Section 1 (the last paragraph).

☒ A4. Have you used AI writing assistants when working on this paper?
Left blank.

B ☑ Did you use or create scientific artifacts?

Section 5

☑ B1. Did you cite the creators of artifacts you used?
Section 5

☑ B2. Did you discuss the license or terms for use and / or distribution of any artifacts?
Section 5. In the footnotes.

☑ B3. Did you discuss if your use of existing artifact(s) was consistent with their intended use, provided that it was specified? For the artifacts you create, do you specify intended use and whether that is compatible with the original access conditions (in particular, derivatives of data accessed for research purposes should not be used outside of research contexts)?
Section 5. In the footnotes.

□ B4. Did you discuss the steps taken to check whether the data that was collected / used contains any information that names or uniquely identifies individual people or offensive content, and the steps taken to protect / anonymize it?
Not applicable. Left blank.

□ B5. Did you provide documentation of the artifacts, e.g., coverage of domains, languages, and linguistic phenomena, demographic groups represented, etc.?
Not applicable. Left blank.

☑ B6. Did you report relevant statistics like the number of examples, details of train / test / dev splits, etc. for the data that you used / created? Even for commonly-used benchmark datasets, include the number of examples in train / validation / test splits, as these provide necessary context for a reader to understand experimental results. For example, small differences in accuracy on large test sets may be significant, while on small test sets they may not be.
Section 5. Table 1.

C ☑ Did you run computational experiments?

Section 5, 6

☑ C1. Did you report the number of parameters in the models used, the total computational budget (e.g., GPU hours), and computing infrastructure used?
Section 6. Table 3, footnote 7.

The Responsible NLP Checklist used at ACL 2023 is adopted from NAACL 2022, with the addition of a question on AI writing assistance.

☑ C2. Did you discuss the experimental setup, including hyperparameter search and best-found hyperparameter values?
Section 5, Table 2.

☑ C3. Did you report descriptive statistics about your results (e.g., error bars around results, summary statistics from sets of experiments), and is it transparent whether you are reporting the max, mean, etc. or just a single run?
Section 6. Table 3

☑ C4. If you used existing packages (e.g., for preprocessing, for normalization, or for evaluation), did you report the implementation, model, and parameter settings used (e.g., NLTK, Spacy, ROUGE, etc.)?
Section 5, 6.

D ☒ Did you use human annotators (e.g., crowdworkers) or research with human participants?

Left blank.

□ D1. Did you report the full text of instructions given to participants, including e.g., screenshots, disclaimers of any risks to participants or annotators, etc.?
No response.

□ D2. Did you report information about how you recruited (e.g., crowdsourcing platform, students) and paid participants, and discuss if such payment is adequate given the participants' demographic (e.g., country of residence)?
No response.

□ D3. Did you discuss whether and how consent was obtained from people whose data you're using/curating? For example, if you collected data via crowdsourcing, did your instructions to crowdworkers explain how the data would be used?
No response.

□ D4. Was the data collection protocol approved (or determined exempt) by an ethics review board?
No response.

□ D5. Did you report the basic demographic and geographic characteristics of the annotator population that is the source of the data?
No response.