Span-level Aspect-based Sentiment Analysis via Table Filling

Mao Zhang[1,2], **Yongxin Zhu**[1,2], **Zhen Liu**[1,2], **Zhimin Bao**[3], **Yunfei Wu**[3]
Xing Sun[3], **Linli Xu**[1,2]
[1]School of Computer Science and Technology, University of Science and Technology of China
[2]State Key Laboratory of Cognitive Intelligence
[3]Tencent YouTu Lab
{zmyyy,zyx2016,liuzhenz}@mail.ustc.edu.cn
{zhiminbao,marcowu,winfredsun}@tencent.com, linlixu@ustc.edu.cn

Abstract

In this paper, we propose a novel span-level model for Aspect-Based Sentiment Analysis (ABSA), which aims at identifying the sentiment polarity of the given aspect. In contrast to conventional ABSA models that focus on modeling the word-level dependencies between an aspect and its corresponding opinion expressions, in this paper, we propose Table Filling BERT (TF-BERT), which considers the consistency of multi-word opinion expressions at the span-level. Specially, we learn the span representations with a table filling method, by constructing an upper triangular table for each sentiment polarity, of which the elements represent the sentiment intensities of the specific sentiment polarity for all spans in the sentence. Two methods are then proposed, including table-decoding and table-aggregation, to filter out target spans or aggregate each table for sentiment polarity classification. In addition, we design a sentiment consistency regularizer to guarantee the sentiment consistency of each span for different sentiment polarities. Experimental results on three benchmarks demonstrate the effectiveness of our proposed model.

1 Introduction

Aspect-based sentiment analysis (Pontiki et al., 2014) ABSA is a fine-grained branch of sentiment analysis, which aims at recognizing the sentiment polarity of the given aspect in the sentence. For example, given the sentence "Boot time is super fast, around anywhere from 35 seconds to 1 minute" and the aspect "Boot time", the opinion expression corresponding to the aspect is "super fast" so that the sentiment polarity of the aspect "Boot time" is positive.

Recently, several methods (Tang et al., 2015; Wang et al., 2016; Ma et al., 2017; Huang et al., 2018; Sun et al., 2019a; Chen et al., 2020; Zhang and Qian, 2020; Xiao et al., 2021; Li et al., 2021; Zhou et al., 2021) have been proposed to exploit the connections between the given aspect and its corresponding opinion expressions in the task of ABSA. Tang et al. (2015) introduces recurrent neural networks (RNNs) to retrieve the aspect-related information by fusing the aspect with its contextualized information in the sentence. Furthermore, Ma et al. (2017); Huang et al. (2018) propose to model the distance dependency between the aspect and the distant opinion expressions with attention mechanisms (Vaswani et al., 2017). To better leverage the syntax information in the ABSA task, some recent studies (Sun et al., 2019a; Xiao et al., 2021; Li et al., 2021) adopt graph neural networks (GNNs) over the dependency trees. Moreover, Chen et al. (2020); Zhou et al. (2021) generate dynamic aspect-specific trees for every sentence-aspect pair to learn the relationships between the aspect words and opinion words.

Despite the improvements achieved by the methods above in the task of ABSA, they take opinion expressions as single words and rely on attention mechanisms to learn the dependency between them, which gives rise to two issues: 1) Word-level dependency ignores the semantics of the entire opinion expressions. 2) Sentiment conflicts may exist in the multi-word opinion expressions since the sentiment polarities predicted over each word can be different (Hu et al., 2019). An example is shown in Figure 1, in which the opinion expression to the aspect "food" is "delicious but expensive". If the model only captures the dependency either between "food" and "delicious" or between "food" and "expensive", it would get the wrong sentiment polarity of positive/negative. Even if all word-level dependencies have been built, the sentiment conflicts between "delicious" and "expensive" may still confuse the model. In principle, if the opinion words "delicious", "but" and "expensive" can be considered simultaneously, it is easier for the model to predict the correct sentiment polarity as neutral.

Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics
Volume 1: Long Papers, pages 9273–9284

To address the above issues, in this paper, we propose a span-level ABSA model and introduce the span-level dependencies, which consider all possible continuous subsequences of a sentence, namely spans, and build connections with the given aspect. While being more flexible, spans are of variable lengths, which inevitably pose significant challenges for standard mechanisms such as attention or GCN. In this paper, we take a different approach with a table filling method to learn span representations naturally and efficiently in the ABSA task, inspired by the success of table filling methods in the relational triple extraction (RTE) task (Zhang et al., 2017; Ren et al., 2021). Based on the span representations, two methods for sentiment polarity classification are introduced, which consist of a table-decoding method and a table-aggregation method. Specifically, we construct an upper triangular table for each sentiment polarity, of which each element represents the sentiment intensity of the specific sentiment polarity for the corresponding span. For the table-decoding method, inspired by Hu et al. (2019), we first select all possible opinion expressions according to the sentiment intensity in the table for each sentiment polarity. Next, we predict the sentiment polarities with the span representations, which are aggregated according to the sentiment intensities of the extracted target spans. For the table-aggregation method, we directly aggregate all sentiment polarity tables to get the probability of the specific sentiment polarity. Additionally, in order to guarantee the sentiment consistency of each span with respect to different sentiment polarities, we design a sentiment consistency regularizer to prevent the same span from getting high sentiment intensities on different tables at the same time.

In summary, the main contributions of the work are as follow:

- To the best of our knowledge, this is the first work to model span-level dependencies between aspects and the corresponding opinion expressions for the ABSA task. We introduce the table filling method and propose our TF-BERT model. We maintain a table for each sentiment polarity, and the elements in the table represent the sentiment intensities of the spans to the given aspect. Moreover, we design a table-decoding method and a table-aggregation method to predict the sentiment polarity.

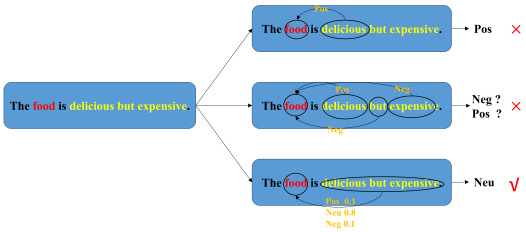


Figure 1: An example sentence of a restaurant review. The aspect words and opinion words are marked in red and yellow, respectively. We denote positive, negative and neutral sentiment as Pos, Neg and Neu, respectively.

- We propose a sentiment consistency regularizer to ensure the sentiment consistency of each span among tables for different sentiment polarities to prevent each span from expressing different sentiments for the given aspect.

- Extensive experimental results on three public standard datasets verify the effectiveness of modeling relationships between aspects and their corresponding opinion expressions in span-level.

2 Related Works

2.1 Aspect-based Sentiment Analysis

The goal of the ABSA task is to identify the sentiment polarity of the given aspect in the sentence (Schouten and Frasincar, 2015; Brauwers and Frasincar, 2021). Earlier methods (Titov and McDonald, 2008; Jiang et al., 2011) based on handcrafted features are not able to build the connections between the aspects and opinion expressions, whose results are largely depending on the quality of features.

To tackle these problems, recent studies focus on using deep learning methods to build the end-to-end models for the ABSA task, which can be categorized into LSTM-based methods, attention-based methods, and GNN-based methods.

LSTM-based Methods LSTM (Hochreiter and Schmidhuber, 1997) is a variant of RNN which is widely used in processing sequential data. Pioneering LSTM-based models treat the sentence as a word sequence and use relatively simple methods to exchange the information between the aspect words and context words. For example, Tang et al. (2015)

aggregates the representations of aspect words to obtain the sentiment representation of the given aspect. However, it is difficult for these methods to deal with the long-distance dependency problem.

Attention-based Methods To model the long distance dependency, Wang et al. (2016); Ma et al. (2017); Huang et al. (2018); Tan et al. (2019) compute the similarity scores between words in a sentence with attention mechanisms. Among them, AOA (Huang et al., 2018) adopts the cross attention from aspect to text and text to aspect simultaneously to model the aspects and sentences jointly to better capture their interactions. To distinguish the conflicting opinions, Tan et al. (2019) combines the positive and negative attention and learns extra aspect embeddings.

GNN-based Methods To better construct the connections between the aspects and the corresponding contexts, a line of works (Sun et al., 2019b; Chen et al., 2020; Zhang and Qian, 2020; Li et al., 2021) leverage the syntactic information by applying GNN on syntax trees. These models regard words in a sentence as nodes in a graph, and learn node representations by aggregating information from adjacent nodes. Therefore, the effect of the distance between aspect words and opinion words is mitigated. Specifically, Sun et al. (2019b) uses GCN over dependency tree to model the sentence structure. Instead of using the vanilla GCN, Zhang and Qian (2020) designs a Bi-level GCN so that the model can assign different attention to different types of edges in a dependency tree. To alleviate the effects of parsing errors and informal expressions, Li et al. (2021) builds an extra semantic graph using the attention mechanism and applies GCN on the syntactic and semantic graph to obtain the aspect-specific representation.

In addition, recent pre-trained models, such as BERT (Devlin et al., 2019), have shown appealing performance in many tasks including ABSA. For instance, by constructing auxiliary sentences, Sun et al. (2019a) converts the ABSA problem into a sentence-pair classification task. Motivated by the neuroscience studies, Zhang et al. (2022) selects the most important word at each step and dynamically changes the aspect-oriented semantics using a dynamic re-weighting adapter.

2.2 Table Filling

Table filling based methods are widely used in RTE task. These methods generate a table for each relation, of which the elements are often used to represent specific information of two entities regarding the given relation, such as start and end positions or entity types. For example, Zhang et al. (2017) maintains an upper triangular table to represent the relations between two words, and fills the table in a specific order. Ren et al. (2021) proposes to mine the global associations of relations and of token pairs using the attention mechanism based on which a proper label is assigned to every item in the table to better construct the table features.

3 Methodology

In this section, we first show the problem definition in Section 3.1, then describe the table filling strategy in Section 3.2, followed by the model details in Section 3.3.

3.1 Problem Definition

In the ABSA task, we are given a sentence-aspect pair (s, a), where $s = \{w_1, w_2, ..., w_n\}$ is a sequence of n words, and $a = \{a_1, a_2, ..., a_m\}$ is an aspect, we denote a span $\{w_i, w_{i+1}, ..., w_j\}$ as $span(i, j)$. The goal of the ABSA task is to precisely predict the sentiment polarity of the given aspect a. In our proposed TF-BERT, we model the relationships between an aspect and its corresponding opinion expressions at the span-level. To effectively handle spans with different lengths, we convert the ABSA task into the task of filling the table for each sentiment polarity so that we can use the start and end positions to denote any span in the same manner.

3.2 Table Filling Strategy

Given the sentence-aspect pair (s, a), we will maintain a table $table_c$ with size $n \times n$ for each sentiment polarity c ($c \in C$, and C contains all distinct sentiment polarities). The $\frac{n\times(n+1)}{2}$ elements in the upper triangular table correspond to the $\frac{n\times(n+1)}{2}$ spans in the sentence s. Unlike the practice in the RTE task, we do not assign a label for each item in the table since there is no supervision information for the table. Instead, we assign each table element a value to represent the sentiment intensity of the corresponding span of the specific sentiment polarity.

3.3 Model

The overall architecture of TF-BERT is shown in Figure 2. It consists of three main modules: an

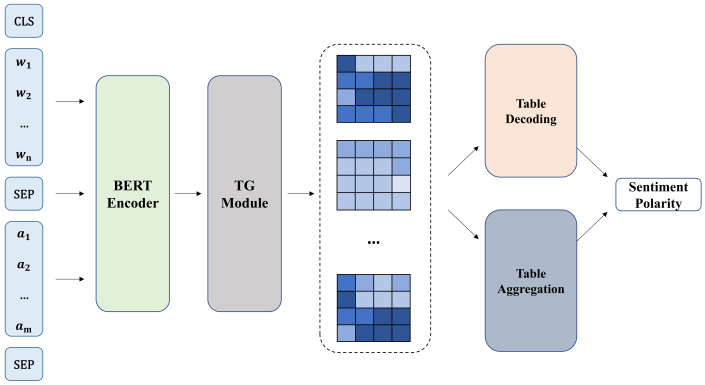


Figure 2: Model architecture

Encoder module, a Table Generation (TG) module, and a Sentiment Classification module.

Encoder We adopt a pre-trained model (i.e., BERT) to map each word in s into a real value vector. Given (s, a), we construct the input as $[\text{CLS}], s, [\text{SEP}], a, [\text{SEP}]$ to obtain the aspect-specific context representations $H = \{h_1, h_2, ..., h_n\}$, where $h_i \in \mathbb{R}^d$.

Then, to make the model aware of the start and end positions of spans, we apply two separated Feed-Forward Networks (FFN) on H to get the initial start and end features, denoted as H_{st} and H_{ed} respectively, which can be formulated as:

$$H_{st} = W_{st}H + b_{st} \quad (1)$$

$$H_{ed} = W_{ed}H + b_{ed} \quad (2)$$

where $W_{st/ed} \in \mathbb{R}^{d\times d}$ and $b_{st/ed} \in \mathbb{R}^d$ are trainable parameters. Then H_{st} and H_{ed} are fed into the Table Generation module.

Table Generation The Table Generation module generates the table for each sentiment polarity. Taking H_{st} and H_{ed} as input we generate the table feature for each span $span(i, j)$ at first, which is denoted as $TF(i, j)$ and computed as follow:

$$TF(i, j) = \sigma(H_{st,i} \otimes H_{ed,j}) \quad (3)$$

where $\otimes$ represents the Hadamard Product operation, σ is the activation function, $H_{st,i}$ and $H_{ed,j}$ are the start and end representations for token w_i and w_j, respectively.

After obtaining the table features, we apply a linear layer for $TF_c(i, j)$ to compute the sentiment intensity of $span(i, j)$ regarding the sentiment polarity c, that is:

$$table_c(i, j) = {W_c}^\top TF(i, j) + b_c \quad (4)$$

where $W_c \in \mathbb{R}^d$ and $b_c \in \mathbb{R}$ are trainable parameters.

Besides, we propose a sentiment consistency regularizer for the generated tables to improve the performance of the Table Generation module. Intuitively, the same span does not show different sentiments for a given aspect. Therefore, we maximize the discrepancy between any two tables for different sentiment polarities to ensure that each span does not get high sentiment intensities on different tables at the same time, which can be formulated as:

$$R_{SC} = \frac{|C|}{\sum_{c\in C}\sum_{c'\in C} ||table_c - table_{c'}||_F} \quad (5)$$

where C is the set of all distinct sentiment polarities.

Sentiment Classification After getting sentiment intensity of a span for each sentiment polarity, we propose two methods to get the sentiment probability distribution to better leverage the table information for sentiment classification, namely table-decoding and table-aggregation.

In the table-decoding method, we design a decoding process to extract the correct opinion expressions for the given aspect. Firstly, we select the spans with the M highest sentiment intensities in every table as the possible opinion expressions and record their start positions S, end positions E and sentiment intensities I. In order to prevent the model from simply choosing too long spans or even the whole sentence as opinion expressions, we propose a span selection algorithm to select K target spans as shown in Algorithm 1. Finally, we use the weighted sum of the corresponding table features

of these selected spans according to their sentiment intensities to get the final sentiment representation h_o, which can be formulated as:

$$S_c, E_c, I_c = \text{top-M}(table_c) \tag{6}$$

$$\begin{aligned} S, E, I = &\text{concat}_{c\in C}(S_c), \\ &\text{concat}_{c\in C}(E_c), \text{concat}_{c\in C}(I_c) \end{aligned} \tag{7}$$

$$O = \text{SpanSelction}(S, E, I, K) \tag{8}$$

$$h_o = \sum_{(S_i,E_i,I_i)\in O} \frac{\exp(I_i)}{\sum_{(S_j,E_j,I_j)\in O}\exp(I_j)} TF(S_i, E_i) \tag{9}$$

Finally, we apply a linear classifier over h_o to compute the sentiment probability distribution, that is:

$$p_{dec} = \text{softmax}(W_o h_o + b_o) \tag{10}$$

where $W_o \in \mathbb{R}^{|C|\times d}$ and $b_o \in \mathbb{R}^{|C|}$ are model parameters.

However, the heuristic decoding algorithm may also select unrelated spans which would introduce noise when predicting the sentiment polarity. Moreover, when M becomes larger, the table-decoding method is time-consuming. In practice, it is not necessary to identify the correct spans, considering that the tables already present the intensities of different sentiment polarities. Therefore, instead of extracting the opinion expressions, we assume that the prediction results should be consistent with the sentiment intensities presented in the tables. In the table-aggregation method, we directly aggregate and concatenate the tables of each sentiment polarity to obtain the sentiment probability distribution, which can be formulated as:

$$table_{\text{agg}} = \text{concat}_{c\in C}(f(table_c)) \tag{11}$$

$$p_{\text{agg}} = \text{softmax}(table_{\text{agg}}) \tag{12}$$

where f represents the aggregating function (i.e., max or mean).

Objective We train the model to minimize the following loss function:

$$\ell(\theta) = -\sum_{i}^{|D|}\sum_{j}^{|C|} y_i^j \log(p_i^j) + \lambda_1||\theta||_2^2 + \lambda_2 R_{SC} \tag{13}$$

where D is the training data set, y_i^j is the ground-truth sentiment polarity, θ represents all trainable model parameters, λ_1 and λ_2 are regularization coefficients, and C denotes all distinct sentiment polarities. The first term represents the standard cross-entropy loss and the second term is L_2-regularization.

Algorithm 1 Span Selection

Input: S, E, I, K
S denotes the start positions of the candidate spans
E denotes the end positions of the candidate spans
I denotes the sentiment intensities of the candidate spans
K is the number of selected spans

```
Let R, O, U = {}, {}, {}
for S_i, E_i, I_i in S, E, I do
    if S_i ≤ E_i then
        r_l = I_i − (E_i − S_i + 1)
        u_l = (S_i, E_i, I_i)
        R = R ∪ {r_l}, U = U ∪ {u_l}
    else
        continue
    end if
end for
while R ≠ {} and |O| < K do
    l = arg max R
    O = O ∪ {u_l}, R = R − {r_l}, U = U − {u_l}
end while
return O
```

Dataset	Division	# Pos.	# Neg.	# Neu.
Laptop	Train	976	851	455
	Test	337	128	167
Restaurant	Train	2164	637	807
	Test	727	196	196
Twitter	Train	1507	1528	3016
	Test	172	169	336

Table 1: Dataset statistics

4 Experiments

4.1 Datasets

We evaluate our proposed TF-BERT model on three benchmark datasets for aspect-based sentiment analysis, including Laptop, Restaurant and Twitter. The Laptop and Restaurant datasets consist of reviews from the SemEval ABSA challenge (Pontiki et al., 2014). The Twitter dataset includes tweets from Dong et al. (2014). We follow Chen et al. (2017) to pre-process these datasets to remove the samples which have conflicting sentiment polarities. Table 1 shows the statistics of the three datasets.

4.2 Implementation Details

We use `bert-base-uncased` to build our framework. The TF-BERT model is trained in 10 epochs with a batch size 16. We use the Adam optimizer with a learning rate 0.00005 for all datasets, and all model weights are initialized with a uniform distribution. The dropout rate is set to 0.3. λ_1 is set to 0.0001 and λ_2 is set to 0.1, 0.3 and 0.15 for the three datasets, respectively. In the table-decoding method, M is set to 5, 7 and 7 for the three datasets, respectively, and K is set to 3 for all

datasets. In the table-aggregation method, we use the mean function to aggregate all the tables and get the probability distribution. All experiments are conducted on a single Nvidia 3090 GPU. We run our model three times with different random seeds and report the average results.

4.3 Baselines

In this subsection, we briefly summarize the baseline models we compare to in the experiments: (1) **ATAE-LSTM** (Wang et al., 2016) combines the attention mechanism with the LSTM network and uses extra aspect embeddings to obtain the aspect-specific representations (2) **IAN** (Ma et al., 2017) employs two LSTMs to model contexts and aspects separately, while using an interactive attention mechanism to exchange information. (3) **AOA** (Huang et al., 2018) uses the aspect-to-text and text-to-aspect cross attention together to introduce interactions between aspect words and context words. (4) **ASGCN** (Zhang et al., 2019) uses GCNs and aspect-aware attention to get the aspect-specific representations. (5) **CDT** (Sun et al., 2019b) utilizes the dependency trees from an external dependency parser to shorten the distance between a given aspect and its corresponding opinion expression, and applies GCNs for information propagation. (6) **DualGCN** (Li et al., 2021) uses both dependency parsing and the attention mechanism to construct syntactic and semantic connections, and exchanges information between them through a mutual Bi-Affine transformation. (7) **BERT** (Devlin et al., 2019) is the vanilla BERT model fine-tuned on the three datasets, and uses the representation of the [CLS] token to build a classifier. (8) **BERT-SPC** (Song et al., 2019) feeds the contexts and aspects into the BERT model for the sentence pair classification task. (9) **RGAT-BERT** (Wang et al., 2020) generates a unified aspect-oriented dependency tree by reshaping and pruning the original dependency tree and proposes a relational graph attention network to encode the tree. (10) **T-GCN**(Tian et al., 2021) designs a type-aware GCN to explicitly utilize the information of dependency types for ABSA. (11) **BERT4GCN** (Xiao et al., 2021) enhances the dependency graph with the attention weights from the intermediate layers in BERT, and apply GCNs over the supplemented dependency graph. (12) **DR-BERT** (Zhang et al., 2022) learns dynamic aspect-oriented semantics with a dynamic re-weighting adapter which selects the most important words at each step and updates the semantics.

4.4 ABSA Results

We use the accuracy and macro-averaged F1-score as the main evaluation metrics. From the results in Table 2, we can first observe that, models using BERT encoders beat most models with LSTM encoders (e.g., ATAE-LSTM, IAN and AOA), which indicates the superiority of the pre-trained language models. In our implementation, we also use the BERT encoder to get the aspect-specific representations. Secondly, the proposed TF-BERT performs better than models using the attention mechanisms and dependency graphs (e.g., CDT, DualGCN and RGAT-BERT), which connect aspect words and opinion words, justifying the effectiveness of TF-BERT to model the dependencies between aspects and the corresponding contexts from the span-level. Concretely, TF-BERT can better understand the semantics of the entire opinion expression and ensure the sentiment consistency for each opinion expression. Moreover, compared with the state-of-the-art baselines (i.e., T-GCN or DR-BERT), our TF-BERT still performs better in both evaluation metrics on the Laptop and Twitter datasets, which demonstrate the effectiveness of the table filling strategy.

4.5 Ablation Study

In this subsection, we conduct ablation studies on the three datasets and further investigate the influence of each component. The results are shown in Table 3. As expected, the full model has the best performance. The model w/o R_{SC} means that we remove the sentiment consistency regularizer, and the performance of TF-BERT drops significantly on all three datasets, which demonstrates that the regularizer can ensure the sentiment consistency for each span across tables for different sentiment polarities. The model w/o separated FFNs means we do not use two separated FFNs to get the initial start and end features. Therefore, the performance degrades substantially on the three datasets which justifies that our TF-BERT is better aware of the start and end positions of every span by using separated FFNs to obtain different start and end features. The model w/o span selection means we directly use the representations of the candidate spans to predict the sentiment polarity in TF-BERT (dec). The results show that our span selection algorithm can help TF-BERT (dec) find the corresponding opinion expressions for the given aspect rather than

Models	Laptop		Restaurant		Twitter	
	Accuracy	Macro-F1	Accuracy	Macro-F1	Accuracy	Macro-F1
ATAE-LSTM (Wang et al., 2016)	68.70	-	77.20	-	-	-
IAN (Ma et al., 2017)	72.10	-	78.60	-	-	-
AOA (Huang et al., 2018)	74.50	-	81.20	-	-	-
ASGCN (Zhang et al., 2019)	75.55	71.05	80.77	72.02	72.15	70.40
CDT (Sun et al., 2019b)	77.19	72.99	82.30	74.02	74.66	73.66
DualGCN (Li et al., 2021)	78.48	74.74	84.27	78.08	75.92	74.29
BERT (Devlin et al., 2019)	77.29	73.36	82.40	73.17	73.42	72.17
BERT-SPC (Song et al., 2019)	78.99	75.03	84.46	76.98	74.13	72.73
RGAT-BERT (Wang et al., 2020)	78.21	74.07	86.60	81.35	76.15	74.88
T-GCN (Tian et al., 2021)	80.88	77.03	86.16	79.95	76.45	75.25
BERT4GCN (Xiao et al., 2021)	77.49	73.01	84.75	77.11	74.73	73.76
DR-BERT (Zhang et al., 2022)	81.45	78.16	**87.72**	**82.31**	77.24	76.10
TF-BERT (dec)	81.49	78.30	86.95	81.43	77.84	76.23
TF-BERT (agg)	**81.80**	**78.46**	87.09	81.15	**78.43**	**77.25**

Table 2: Performance comparison on three benchmark datasets. The best scores are bolded, and the second best ones are underlined.

Models	Laptop		Restaurant		Twitter	
	Accuracy	Macro-F1	Accuracy	Macro-F1	Accuracy	Macro-F1
TF-BERT (dec)	**81.49**	**78.30**	**86.95**	**81.43**	**77.84**	**76.23**
w/o R_{SC}	80.85	77.27	85.97	79.71	75.78	74.50
w/o separated FFNs	80.38	76.03	85.79	78.74	76.66	76.01
w/o span selection	80.54	77.17	86.15	79.97	75.48	74.75
TF-BERT (agg)	**81.80**	**78.46**	**87.09**	**81.15**	**78.43**	**77.25**
w/o R_{SC}	80.70	77.43	86.60	80.72	76.51	75.45
w/o separated FFNs	81.17	77.94	85.70	78.62	77.10	75.35

Table 3: Ablation study on three benchmark datasets.

#	Reviews	IAN	TF-BERT	target spans
1	*Set up* was *easy*.	Pos ✔	Pos ✔	"easy","."
2	*Did not enjoy* the new *Windows 8* and touchscreen functions.	Neu ✖	Neg ✔	"did not enjoy","not","did"
3	Works well, and I am *extremely happy* to be back to an *apple OS*.	Pos ✔	Pos ✔	"extremely happy","extremely","happy"

Table 4: Comparison of the selected opinion expressions between human and TF-BERT. Aspect and opinion words are in italic. The duplicate spans selected by TF-BERT are removed. We denote positive, negative and neutral sentiment as Pos, Neg and Neu, respectively.

Models	Laptop		Restaurant		Twitter	
	Accuracy	Macro-F1	Accuracy	Macro-F1	Accuracy	Macro-F1
Att+GCN	79.11	75.53	85.52	77.55	76.81	75.07
Dep+GCN	**80.22**	**77.04**	85.88	79.52	76.51	75.33
Span	79.43	75.83	**86.06**	**80.21**	**76.96**	**75.55**

Table 5: Performance comparison of models using different kinds of features on on three benchmark datasets.

Models	Parameter Number	Laptop		Restaurant		Twitter	
		T	E	T	E	T	E
TF-BERT (dec)	110.6M	98s	10	222s	10	364s	10
TF-BERT (agg)	110.6M	80s	10	196s	10	233s	10
DR-BERT* (Zhang et al., 2022)	-	157s	10	183s	10	379s	10
DualGCN-BERT (Li et al., 2021)	111.8M	100s	15	276s	15	293s	15

Table 6: Computation runtime on three benchmark datasets. "T" and "E" represent the training time of each epoch (seconds) and the number of training epochs required, respectively. "*" means that we report the results shown in the original paper. For other models, we conduct experiments on a single Nvidia 3090 GPU with the same batch size 16 and report the results.

simply choosing the too long spans.

4.6 Case Study

To investigate whether the proposed TF-BERT can correctly figure out the complex opinion expressions for the given aspects, we select a few sample cases and present the predictions and target opinion expressions extracted by Algorithm 1. The results are shown in Table 4. First, we can observe that, for the given aspects, the corresponding opinion expressions are among the selected target spans and our TF-BERT can make right sentiment predictions. These results demonstrate that TF-BERT can correctly construct the connections between the given aspects and corresponding opinion expressions and understand the semantics of the entire opinion expressions. Second, even in the complex scenarios when there are multiple aspects in the sentence, TF-BERT can still accurately distinguish the opinion expressions corresponding to each aspect. For example, for the aspect "Windows 8" in the review "Did not enjoy the new Windows 8 and touchscreen functions", the opinion expression "did not enjoy" is selected by TF-BERT while IAN does not capture the key words "did not". In summary, these three examples demonstrate the proposed TF-BERT, by modeling the dependencies between aspects and opinion expressions from the span-level, can connect the opinion expressions with the given aspects through table filling.

4.7 Analysis on the Span-level Features

To better demonstrate the effectiveness of using span features in the ABSA task, we implement the following two models based on word-level dependencies: (1) **Att + GCN** uses the attention mechanism to build connections between each pair of words and applies GCN on the attention weight matrix, (2) **Dep + GCN** utilizes the dependency parse graph to connect aspect words and opinion words, and applies GCN over the graph. Both models are based on the BERT encoder and use the corresponding word features of the given aspect to predict the sentiment polarity.

We compare these two word-level models with a simplified variant (**Span**) of the proposed TF-BERT which directly uses the table features for the given aspect to predict the sentiment polarity. The results are shown in Table 5, where we can observe that, with only simple FFNs to obtain the span representations, **Span** consistently outperforms **Att+GCN** and **Dep+GCN** on almost all datasets, which justify that treating the opinion expressions as spans rather than single words can help better understand the semantics and ensure the sentiment consistency of the entire opinion expressions.

4.8 Analysis on the Computational Cost

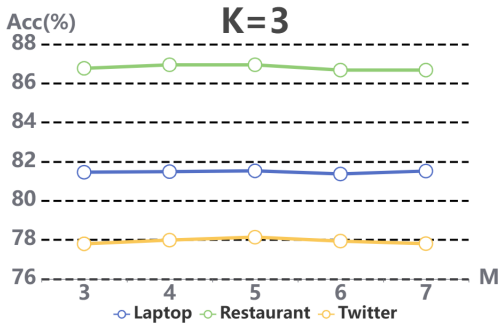
Theoretically, the number of spans in a sentence of length n is $\frac{n\times(n+1)}{2}$, and we need to consider all spans and generate a table for every sentiment polarity $c \in C$, which leads to the time complexity of $O(|C|n^2)$ to fill out all tables. Empirically, to investigate the computational costs of the proposed TF-BERT, we compare the running time and number of trainable parameters of TF-BERT with two baseline methods. As shown in Table 6, compared to other BERT-based baseline models, TF-BERT takes less training time in each epoch with comparable model size, which demonstrate that our TF-BERT model does not incur extra computational costs.

4.9 Effects of Hyper-Parameters

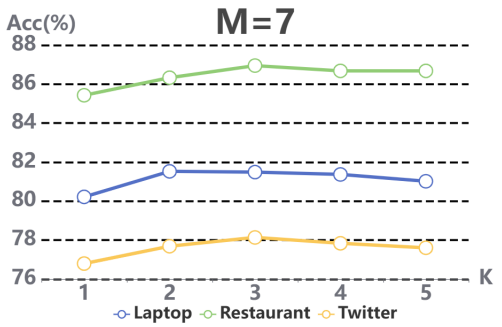
To investigate the impact of the hyper-parameters M and K in the table-decoding method, we evaluate our model on the three datasets by fixing the value of one of them and varying the other. As shown in Figure 3(a), for a fixed K, the table-decoding method is robust to the number of candidate spans. Meanwhile, although a larger K does improve the accuracy, it also introduces additional noise. Setting K to around 3 leads to consistently better performance.

5 Conclusion

In this paper, we propose a novel table filling based model TF-BERT for the ABSA task, which maintains an upper triangular table for each sentiment polarity and the elements in the table denotes the sentiment intensity of the specific sentiment polarity for all spans in the sentence. Specifically, we first append the given aspect to the sentence and use the BERT model to encode the augmented sentence to get the aspect-specific representations. Then, we construct the span features and generate a table for each sentiment polarity. Finally, we utilize two methods to obtain the sentiment probability distribution. Additionally, to ensure the sentiment consistency of the same span across different tables, we adopt a sentiment consistency regularizer on the generated tables. Extensive experiments on



(a) Impact of M.



(b) Impact of K.

Figure 3: Accuracy on the three datasets with different hyper-parameter settings.

three benchmarks demonstrate the effectiveness of our TF-BERT model.

6 Limitations

First, our method needs to check all spans in the given sentence and build a table for each sentiment polarity, and is therefore difficult to handle too long sentences. Another limitation of our work is that for the different aspects in the same sentence, we need to rebuild the tables.

7 Acknowledgments

This research was supported by the National Key Research and Development Program of China (Grant No. 2022YFB3103100), the National Natural Science Foundation of China (Grant No. 62276245), and Anhui Provincial Natural Science Foundation (Grant No. 2008085J31).

References

Gianni Brauwers and Flavius Frasincar. 2021. A survey on aspect-based sentiment classification. *ACM Computing Surveys*, 55:1 – 37.

C. Chen, Z. Teng, and Y. 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)*.

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

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, Minneapolis, Minnesota. Association for Computational Linguistics.

Li Dong, Furu Wei, Chuanqi Tan, Duyu Tang, and Ke Xu. 2014. Adaptive recursive neural network for target-dependent twitter sentiment classification. In *Meeting of the Association for Computational Linguistics*.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9:1735–80.

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, Florence, Italy. Association for Computational Linguistics.

Binxuan Huang, Yanglan Ou, and Kathleen M. Carley. 2018. Aspect level sentiment classification with attention-over-attention neural networks. *Springer, Cham*.

Long Jiang, Mo Yu, Ming Zhou, Xiaohua Liu, and Tiejun Zhao. 2011. Target-dependent Twitter sentiment classification. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 151–160, Portland, Oregon, USA. Association for Computational Linguistics.

Ruifan Li, Hao Chen, Fangxiang Feng, Zhanyu Ma, Xiaojie Wang, and Eduard Hovy. 2021. Dual graph convolutional networks for aspect-based sentiment analysis. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 6319–6329, Online. Association for Computational Linguistics.

Dehong Ma, Sujian Li, Xiaodong Zhang, and Houfeng Wang. 2017. Interactive attention networks for aspect-level sentiment classification. *ArXiv*, abs/1709.00893.

M. Pontiki, D Galanis, J. Pavlopoulos, H. Papageorgiou, and S. Manandhar. 2014. Semeval-2014 task 4: Aspect based sentiment analysis. *Proceedings of International Workshop on Semantic Evaluation at*.

Feiliang Ren, Longhui Zhang, Shujuan Yin, Xiaofeng Zhao, Shilei Liu, Bochao Li, and Yaduo Liu. 2021. A novel global feature-oriented relational triple extraction model based on table filling. In *EMNLP*.

Kim Schouten and Flavius Frasincar. 2015. Survey on aspect-level sentiment analysis. *IEEE Transactions on Knowledge and Data Engineering*, 28:1–1.

Youwei Song, Jiahai Wang, Tao Jiang, Zhiyue Liu, and Yanghui Rao. 2019. Attentional encoder network for targeted sentiment classification. *ArXiv*, abs/1902.09314.

Chi Sun, Luyao Huang, and Xipeng Qiu. 2019a. Utilizing BERT for aspect-based sentiment analysis via constructing auxiliary sentence. 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 380–385, Minneapolis, Minnesota. Association for Computational Linguistics.

K. Sun, R. Zhang, S. Mensah, Y Mao, and X. Liu. 2019b. 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)*.

X. Tan, Y. Cai, and C. Zhu. 2019. Recognizing conflict opinions in aspect-level sentiment classification with dual 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)*.

D Tang, B. Qin, X. Feng, and T. Liu. 2015. Effective lstms for target-dependent sentiment classification. *Computer Science*.

Yuanhe Tian, Guimin Chen, and Yan Song. 2021. 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. Association for Computational Linguistics.

Ivan Titov and Ryan McDonald. 2008. Modeling online reviews with multi-grain topic models. In *Proceedings of the 17th International Conference on World Wide Web*, WWW '08, page 111–120, New York, NY, USA. Association for Computing Machinery.

Ashish Vaswani, Noam M. Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *NIPS*.

Kai Wang, Weizhou Shen, Yunyi Yang, Xiaojun Quan, and Rui Wang. 2020. Relational graph attention network for aspect-based sentiment analysis. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 3229–3238, Online. Association for Computational Linguistics.

Yequan Wang, Minlie Huang, Xiaoyan Zhu, and Li Zhao. 2016. Attention-based lstm for aspect-level sentiment classification. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*.

Zeguan Xiao, Jiarun Wu, Qingliang Chen, and Congjian Deng. 2021. BERT4GCN: Using BERT intermediate layers to augment GCN for aspect-based sentiment classification. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 9193–9200, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Chen Zhang, Qiuchi Li, and Dawei Song. 2019. Aspect-based sentiment classification with aspect-specific graph convolutional networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4568–4578, Hong Kong, China. Association for Computational Linguistics.

Kai Zhang, Kun Zhang, Mengdi Zhang, Hongke Zhao, Qi Liu, Wei Wu, and Enhong Chen. 2022. Incorporating dynamic semantics into pre-trained language model for aspect-based sentiment analysis. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 3599–3610, Dublin, Ireland. Association for Computational Linguistics.

M. Zhang, Z. Yue, and G. Fu. 2017. End-to-end neural relation extraction with global optimization. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*.

Mi Zhang and Tieyun Qian. 2020. Convolution over hierarchical syntactic and lexical graphs for aspect level sentiment analysis. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 3540–3549, Online. Association for Computational Linguistics.

Yuxiang Zhou, Lejian Liao, Yang Gao, Zhanming Jie, and Wei Lu. 2021. To be closer: Learning to link up aspects with opinions. In *EMNLP*.

ACL 2023 Responsible NLP Checklist

A For every submission:

☑ A1. Did you describe the limitations of your work?
section 6

☐ A2. Did you discuss any potential risks of your work?
Not applicable. Left blank.

☑ A3. Do the abstract and introduction summarize the paper's main claims?
abstract section 1

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

B ☑ Did you use or create scientific artifacts?

section 4

☑ B1. Did you cite the creators of artifacts you used?
section 4

☑ B2. Did you discuss the license or terms for use and / or distribution of any artifacts?
section 4

☑ 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 4

☑ 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?
section 4

☑ B5. Did you provide documentation of the artifacts, e.g., coverage of domains, languages, and linguistic phenomena, demographic groups represented, etc.?
section 4

☑ 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 4

C ☑ Did you run computational experiments?

section 4.8

☑ 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 4.8

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 4.2 4.8

☑ 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 4.2

☑ 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 4.2

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.