Label Correction Model for Aspect-based Sentiment Analysis

Qianlong Wang **Jiangtao Ren***
School of Data and Computer Science
Guangdong Province Key Lab of Computational Science
Sun Yat-sen University, China
wangqlong3@mail3.sysu.edu.cn, issrjt@mail.sysu.edu.cn

Abstract

Aspect-based sentiment analysis includes opinion aspect extraction and aspect sentiment classification. Researchers have attempted to discover the relationship between these two sub-tasks and have proposed the joint model for solving aspect-based sentiment analysis. However, they ignore a phenomenon: aspect boundary label and sentiment label of the same word can correct each other. To exploit this phenomenon, we propose a novel deep learning model named the label correction model. Specifically, given an input sentence, our model first predicts the aspect boundary label sequence and sentiment label sequence, then re-predicts the aspect boundary (sentiment) label sequence using the embeddings of the previously predicted sentiment (aspect boundary) label. The goal of the re-prediction operation (can be repeated multiple times) is to use the information of the sentiment (aspect boundary) label to correct the wrong aspect boundary (sentiment) label. Moreover, we explore two ways of using label embeddings: add and gate mechanism. We evaluate our model on three benchmark datasets. Experimental results verify that our model achieves state-of-the-art performance compared with several baselines.

1 Introduction

Aspect-based sentiment analysis (ABSA) aims to extract the opinion aspects mentioned in the sentence and to predict the sentiment of each aspect (Pontiki et al., 2014). For example, in the review "*Average to good thai food , but terrible delivery .*", *thai food* and *delivery* are aspects, with *positive* and *negative* as their corresponding sentiments.

In the literature, ABSA is usually divided into two sub-tasks, namely, opinion aspect extraction and aspect sentiment classification. The goal of opinion aspect extraction is to extract all aspects present in a sentence. Early work (Zhuang et al., 2006) focuses on detecting the pre-defined aspects in a sentence. Then, some work (Jakob and Gurevych, 2010; Liu et al., 2015) regards the opinion aspect extraction as a sequence labeling task. The goal of aspect sentiment classification is to predict the sentiment of the given opinion aspect, and this sub-task has drawn growing research interests over the past few years (Ma et al., 2017; Wang et al., 2018; Li et al., 2019b). However, most previous work focuses on only one of the sub-tasks. This limits the practical application of ABSA. To apply the existing methods of two sub-tasks in practical applications (i.e., not only extracting aspects but also predicting their sentiments), the most common way is to pipeline the methods of two sub-tasks together.

To further promote the resolution of ABSA, researchers have attempted to discover the relationship between these two sub-tasks and have developed the joint models (Mitchell et al., 2013; Zhang et al., 2015). They utilize a set of aspect boundary labels (e.g., B, I, E, S, O) and a set of sentiment labels (e.g., POS, NEG, NEU and O) to make the models of two sub-tasks jointly trained. In other words, ABSA is modeled as an extension to opinion aspect extraction, where an extra sentiment label is assigned to each word, in addition to the aspect boundary label. Table 1 gives an example of the labeling scheme in the

*Corresponding author.

Proceedings of the 28th International Conference on Computational Linguistics, pages 822–832
Barcelona, Spain (Online), December 8-13, 2020

	Average	to	good	thai	food	,	but	terrible	delivery	.
aspect boundary label	O	O	O	B	E	O	O	O	S	O
sentiment label	O	O	O	POS	POS	O	O	O	NEG	O

Table 1: Labelling scheme used in the joint model, where POS and NEG represent *positive* and *negative* emotions, respectively.

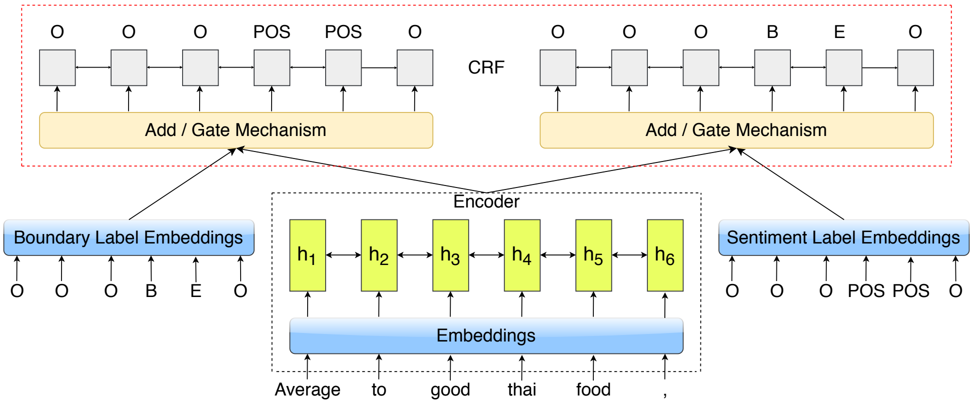

O O O POS POS O
CRF
O O O B E O
Add / Gate Mechanism
Add / Gate Mechanism
Encoder
Boundary Label Embeddings
O O O B E O
h_1 h_2 h_3 h_4 h_5 h_6
Embeddings
Average to good thai food ,
Sentiment Label Embeddings
O O O POS POS O

Figure 1: Overview of the proposed model. The red dotted rectangular part can be run repeatedly during testing, but only once during training. (best viewed in color)

joint model. Although the joint model presents competitive results on the ABSA, researchers overlook a phenomenon: aspect boundary label and sentiment label of the same word can correct each other. Refer to the example in Table 1, if the sentiment label of word *food* is O, we can correct it based on the obvious fact: O cannot be the sentiment label of an aspect word. Therefore, we can reduce the error space in predicting the sentiment label by considering the corresponding aspect boundary label, vice versa.

To make use of this phenomenon to assist ABSA, in this paper, we propose a novel neural networks model named the label correction model. Figure 1 shows the architecture of our model. For a joint model solution to ABSA, each word in the input sentence has two labels (i.e., aspect boundary label and sentiment label), thus our motivation is to use one label to correct the other as much as possible. During training, our model utilizes the ground truth label sequence to improve performance directly. In particular, our model uses the ground truth sentiment label embeddings and hidden states to predict the aspect boundary label sequence. For the prediction of the sentiment label sequence, the operation is similar. During testing, we first predict the aspect boundary label sequence and sentiment label sequence of the input sentence, and such predictions are based solely on hidden states. Then, these two label sequences are employed as features for predictions for the next round. Such design is based on the assumption that in the process of prediction, the proposed model has the ability to correct the label by adopting another label as a feature. Note that the re-prediction operation can be repeated many times. In this work, we try two different ways to use label embeddings: add and gate mechanism. The former integrates the label embeddings into the hidden states by adding. The latter consisting of a sigmoid layer and a pointwise multiplication operation is a way to optionally let label embeddings information through.

To summarize, we make the following contributions in this paper:

- We manifest a phenomenon ignored by researchers that aspect boundary label and sentiment label of the same word can correct each other in the joint model of ABSA.

- We propose a novel model utilizing this phenomenon to solve ABSA. To the best of our knowledge, our model is the first work related to this phenomenon.

- Experimental results on three benchmark datasets show that our model outperforms the strong joint baselines. We conduct an ablation study to quantitatively demonstrate the effectiveness of exploiting this phenomenon.

2 Related Work

Aspect-based sentiment analysis is usually divided into two sub-tasks, namely, opinion aspect extraction and aspect sentiment classification.

Opinion Aspect Extraction Opinion aspect extraction is a fundamental task in ABSA and aims at extracting all aspects present in a sentence. Hu and Liu (2004) first proposed to evaluate the sentiment of different aspects in an opinion sentence, and all aspects are predefined artificially. However, the manually defined aspects cannot cover all aspects appearing in a sentence. Therefore, many researchers turn to extract all possible aspects in a sentence and model the opinion aspect extraction as a sequence labeling problem. They initially used traditional machine learning (Jin and Ho, 2009; Jakob and Gurevych, 2010; Liu et al., 2013) to solve this sub-task. With the developments of deep learning, neural networks based methods (Liu et al., 2015; Wang et al., 2016; Wang et al., 2017; Xu et al., 2018) have achieved better performance on opinion aspect extraction.

Aspect Sentiment Classification Aspect sentiment classification is often interpreted as a multi-class classification problem, assuming that the aspects are given. Traditional approaches usually first manually build a set of features and then run them through SVM (Jiang et al., 2011; Wagner et al., 2014). The feature-based models depend on the quality of laborious feature engineering work and are labor-intensive. Therefore, recent work mainly focuses on capturing the interaction between the aspect and the sentence, by utilizing various neural architectures such as LSTM (Tang et al., 2016a) with attention mechanism (Li et al., 2018a), CNN (Xue and Li, 2018), Memory Networks (Tang et al., 2016b; Chen et al., 2017), and pre-trained models (Wang and Ren, 2020).

Aspect-based Sentiment Analysis Aspect-based sentiment analysis needs to directly predict the sentiment towards an aspect along with discovering the aspect itself. In addition to the pipeline methods, previous work attempted to discover the relationship between its two sub-tasks and presented a more integrated solution to solve ABSA. Specifically, Mitchell et al. (2013) formulated ABSA as a sequence labeling problem and proposed to use CRF with hand-crafted linguistic features. Zhang et al. (2015) leveraged the linguistic features and word embeddings to further improve the performance of the CRF based model. Recently, Li et al. (2019a) proposed a unified model, which contains two stacked LSTMs along with carefully-designed components for maintaining sentiment consistency and improving aspect detection, and achieved state-of-the-art results. However, researchers overlooked a phenomenon: aspect boundary label and sentiment label of the same word can correct each other. In this paper, we further improve performance on ABSA by taking advantage of this phenomenon.

3 Proposed Model

3.1 Problem Formulation

We formulate the complete ABSA as two sequence labeling problems and employ a set of aspect boundary labels $\mathcal{Y}^{\mathcal{B}}$ and a set of sentiment labels $\mathcal{Y}^{\mathcal{S}}$ to accomplish opinion aspect extraction sub-task and aspect sentiment classification sub-task, respectively. Here, $\mathcal{Y}^{\mathcal{B}} = \{\mathrm{B, I, E, S, O}\}$ (short for beginning, inside, ending, single token, outside) and $\mathcal{Y}^{\mathcal{S}} = \{\mathrm{POS, NEG, NEU, O}\}$ (short for positive, negative, neutral, outside). For a given input sequence $\mathrm{X} = \{x_1, ..., x_T\}$ with length T, the goal of our model is to predict two label sequences $\mathrm{Y}^B = \{y_1^B, ..., y_T^B\}$ and $\mathrm{Y}^S = \{y_1^S, ..., y_T^S\}$, where $y_i^B \in \mathcal{Y}^{\mathcal{B}}$ and $y_i^S \in \mathcal{Y}^{\mathcal{S}}$. Based on these two label sequences, we can obtain the extracted aspects along with corresponding sentiments. Taking the sentence in Table 1 as an example, the extracted aspects are *thai food* and *delivery*, whose sentiments are *positive* and *negative*, respectively.

3.2 Overview

Figure 1 shows the architecture of our model. Our model consists of three components, which are the encoder, the add and gate mechanism, and the CRF. The encoder is BERT[1] (Devlin et al., 2019)

[1] In this paper, BERT refers to $\mathrm{BERT_{BASE}}$.

which maps word embeddings into contextualized hidden states using the pre-trained transformer blocks (Vaswani et al., 2017). Then, the add or gate mechanism is utilized to fuse the hidden states and the label embeddings, aiming at using one label's information to enhance the prediction of another. Note that we use the ground truth label sequence during training but label sequence from the latest round of prediction during testing. Finally, to predict the sentiment label sequence and aspect boundary label sequence, the two outputs are fed into two CRFs, respectively.

3.3 Encoder

We use BERT (Devlin et al., 2019) as our text encoder. In this work, we first tokenize the sentence X using WordPiece embeddings (Wu et al., 2016) with a 30,522 token vocabulary and then generate the input sequence $\overline{\mathrm{X}}$ by concatenating a [CLS] token, the tokenized sentence, and a [SEP] token. Then for each token $\overline{x}_i$ in $\overline{\mathrm{X}}$, we convert it into vector space by summing the token, segment, and position embeddings, thus yielding the input embeddings $\mathrm{H}^0 \in \mathbb{R}^{(T+2)\times d}$, where d is the hidden state dimension. Next, we use L stacked transformer blocks to project the input embeddings into a sequence of contextual hidden states $\mathrm{H}^i \in \mathbb{R}^{(T+2)\times d}$ as:

$$\mathrm{H}^i = \text{TransformerBlock}(\mathrm{H}^{i-1}) \quad \forall i \in [1, L] \tag{1}$$

Here, we omit an exhaustive description of the transformer block and refer readers to Vaswani et al. (2017) for more details.

3.4 Add and Gate Mechanism

As the example in Table 1 shows, each word in the input sentence has two labels (i.e., aspect boundary label and sentiment label) in the joint model. Our motivation is to use one label to correct the other. To input labels as features, we first convert each label in two label sequences (Y^B and Y^S) [2] into vector space, so producing the boundary label embeddings $\mathrm{H}^B \in \mathbb{R}^{(T+2)\times d}$ and the sentiment label embeddings $\mathrm{H}^S \in \mathbb{R}^{(T+2)\times d}$. We then propose two ways to use the label embeddings.

Add As the name suggests, the add is a pretty simple way to incorporate the label embeddings directly into the hidden states:

$$\mathrm{H}^{LB(LS)} = \mathrm{H}^L + \mathrm{H}^{B(S)} \tag{2}$$

Although this way is uncomplicated, it is not very effective. When our model predicts the sentiment labels, it considers the hidden states and the boundary label embeddings to be equally important, and vice versa. However, this may not be the case. Because the contextualized hidden states may play a greater role when predicting the sentiment labels for certain words. Taking the sentence in Table 1 as an example, when predicting the sentiment label of word *delivery*, the contextual information containing word *terrible* is significantly more useful than its own boundary label S. Therefore, we propose the gate mechanism to solve the shortcoming of this way.

Gate Mechanism For label prediction of different words, the contextual hidden states and the label embeddings should play significantly different roles. Inspired by gates in LSTM, we propose the gate mechanism to select important information from the hidden states and the label embeddings to forecast label. The gate mechanism can construct tailored hidden states by considering the label embeddings. In detail, it takes the hidden states H^L and the label embeddings $\mathrm{H}^{B(S)}$ as input and outputs a gate matrix $\mathrm{G}^{B(S)}$ to select H^L and $\mathrm{H}^{B(S)}$:

$$\mathrm{G}^{B(S)} = \sigma(\mathrm{H}^L * \mathrm{H}^{B(S)}) \tag{3}$$

$$\mathrm{H}^{LB(LS)} = \mathrm{H}^L \odot \mathrm{G}^{B(S)} + \mathrm{H}^{B(S)} \odot (1 - \mathrm{G}^{B(S)}) \tag{4}$$

[2] In the training stage, we map the ground truth labels into vectors. However, in the inference stage, we map the labels of the latest round of prediction into vectors. As for the first round of prediction, we set the label embeddings to **0**.

Datasets	Train			Dev			Test			Total		
	#POS	#NEG	#NEU	#POS	#NEG	#NEU	#POS	#NEG	#NEU	#POS	#NEG	#NEU
Laptop	898	787	426	96	83	38	341	128	169	1335	998	633
Rest	2839	967	624	218	81	45	1516	494	260	4573	1542	929
Twitter	–	–	–	–	–	–	–	–	–	692	263	2244

Table 2: Dataset statistics. #POS, #NEG and #NEU denote the number of *positive*, *negative* and *neutral* emotions, respectively. Note that there is no standard train-test split on Twitter.

where σ denotes the sigmoid activation function and $\odot$ is element-wise multiplication. From Equation 3, we can find $\mathrm{G}^{B(S)} \in \mathbb{R}^{(T+2)\times d}$ is a matrix whose values are between 0 and 1. Therefore, we can utilize $||\mathrm{G}_i^{B(S)}||_2$[3] to measure the degree of the filter and call it *2-Norm Gate* value. A high value means most of the information in H_i^L is passed from the filter, which results in predicting labels with more contextual information. A low value allows the label embeddings to pass through the filter. This illustrates that using only contextual information is not sufficient, and the information of the other label can assist and correct the prediction of the label.

3.5 Conditional Random Field

It has been shown that CRF (Lafferty et al., 2001) can produce higher labeling accuracy in sequence labeling tasks because it considers the correlations between labels in neighborhoods. Therefore, instead of decoding each label independently with the softmax layer, we predict the sentiment label sequence and aspect boundary label sequence by using two CRFs. In the training stage, we maximize the log-likelihood function:

$$\mathcal{L} = P(\mathrm{Y}^B|\mathrm{X}) + P(\mathrm{Y}^S|\mathrm{X}) \tag{5}$$

where the label sequence likelihood $P(\mathrm{Y}^{B(S)}|\mathrm{X})$ can be computed by the softmax equation. In the test stage, the Viterbi algorithm (Forney, 1973) is used to output the optimized label sequence.

4 Experiments

4.1 Setup

Datasets We conduct experiments on the three benchmark ABSA datasets. Table 2 gives the statistics of three datasets. Laptop contains product reviews from the laptop domain and the train-test split is the same as the original dataset (Pontiki et al., 2014). Rest is the union set of the restaurant domain from SemEval 2014 (Pontiki et al., 2014), 2015 (Pontiki et al., 2015), and 2016 (Pontiki et al., 2016). The new training dataset[4] is obtained by merging three years' training set and the new testing set is built in the same way. Twitter is built by Mitchell et al. (2013), which consists of twitter posts. For Laptop and Rest, we regard 10% randomly held-out training data as the development set. For Twitter, we follow previous work (Zhang et al., 2015; Li et al., 2019a) and report the ten-fold cross-validation results as there is no standard train-test split.

Model Settings We use the publicly available BERT's code[5] to implement our encoder. The hyper-parameters (e.g., word pieces vocabulary size, hidden size of encoder, and learning rate) and optimizer of our model are the same as that of BERT, and we employ the uncased pre-trained model to initialize our WordPiece embeddings and encoder's parameters. We set the dimensions of boundary label embeddings and sentiment label embeddings to 768, and these two label embeddings are randomly initialized from $\mathcal{N}(0,1)$ and learned from scratch. The batch size is 16 and a dropout (Srivastava et al., 2014) probability of 0.1 is used to avoid overfitting. For the maximum length of the input sentence after WordPiece tokenization, we set it to 256. For the gold labels of [CLS] and [SEP] token in the input sequence, we set them to O.

[3] $\mathrm{G}_i^{B(S)} \in \mathbb{R}^d$ is the i-th vector of $\mathrm{G}^{B(S)}$.

[4] Since the training set of SemEval 2016 contains the test examples of SemEval 2015, we exclude these test examples in the training set.

[5] `https://github.com/huggingface/transformers`

Metrices We adopt the precision (P), recall (R), and F1 score as the evaluation metrics. We can obtain the extracted aspects and the corresponding sentiments from the aspect boundary label sequence and the sentiment label sequence, respectively. An extracted aspect is considered to be correct only if it exactly matches the gold aspect and its corresponding sentiment is the same as the gold sentiment label.

4.2 Baselines

Models solving ABSA can generally be divided into three categories: **pipeline**, **joint**, and **collapsed**. The **pipeline** model first detects aspects from the input text and then predicts sentiments over aspects. The **joint** model jointly extracts aspects and predicts their sentiments using a set of aspect boundary labels as well as a set of sentiment labels. The **collapsed** model uses a set of collapsed labels (e.g., B-POS) to directly indicate the boundary of targeted sentiment.

We compare our model with the following methods:

- **CRF-{pipeline, joint, collapsed}**[6] (Mitchell et al., 2013) are three kinds of approaches under the Conditional Random Fields framework.

- **NN-CRF-{pipeline, joint, collapsed}**[7] (Zhang et al., 2015) enhance the CRF framework by introducing a fully connected layer to consolidate the linguistic features and word embeddings.

- **LSTM-CRF**[8] (Lample et al., 2016) consists of LSTM and CRF without feature engineering. We consider it as a collapsed model and run the officially released code to produce results.

- **HAST-TNet**[9] is the pipeline approach of HAST(Li et al., 2018b) and TNet (Li et al., 2018a), which are the current state-of-the-art models on the tasks of aspect boundary detection and aspect sentiment classification respectively. We use the officially released codes to produce the results.

- **UNIFIED** (Li et al., 2019a) is the current state-of-the-art model on ABSA. This collapsed model enhanced with multi-task learning contains two stacked LSTMs.

- **BERT-CLS** treats ABSA as a multi-class classification task, classifying each token in the input text into one of the collapsed labels.

- **BERT-CRF-{joint, collapsed}** is similar to **NN-CRF**. The difference is that we employ BERT to extract representation features instead of word embeddings and linear layers.

4.3 Results

For simplicity, we denote our label correction model as **LCM**. Here, **LCM-Add** adds the label embeddings to the hidden states, and **LCM-Gate** uses the gate mechanism to optionally let the label embeddings information through.

Main Results Table 3 shows our comparisons with baselines on ABSA. The experimental results suggest that our model consistently exhibits the best F1 scores on the three datasets and significantly outperforms the baselines.

Five main observations can be obtained from Table 3. First, although NN-CRF using the word embeddings defeats CRF, it is not strong. Consequently, we add BERT-CRF as another baseline. As shown in Table 3, BERT-CRF achieves much better results than {NN, LSTN}-CRF on all datasets. This indicates that the contextual representations produced by BERT for each token are quite powerful, which is why we chose BERT as the encoder. Second, compared with HAST-TNet, the pipeline of two state-of-the-art models, our model achieves 6.57%, 3.85%, and 4.63% absolute gains on Laptop, Rest, and Twitter respectively, suggesting that a carefully-designed joint model can be more effective than the pipeline

[6] `http://www.m-mitchell.com/code/index.html`

[7] `https://github.com/SUTDNLP/NNTargetedSentiment`

[8] `https://github.com/glample/tagger`

[9] Available at: `https://github.com/lixin4ever/HAST` and `https://github.com/lixin4ever/TNet` respectively

Models	Laptop			Rest			Twitter		
	P	R	F1	P	R	F1	P	R	F1
CRF-pipeline†	59.73	47.56	52.96	52.46	51.22	51.83	42.90	26.26	32.57
CRF-joint†	57.76	36.89	45.02	60.43	49.25	54.27	43.92	24.97	31.84
CRF-collapsed†	60.91	41.45	49.33	64.14	58.02	60.92	48.98	20.21	28.61
NN-CRF-pipeline♮	57.72	49.32	53.19	60.09	61.93	61.00	43.71	37.12	40.06
NN-CRF-joint♮	55.64	34.48	45.49	61.56	50.00	55.18	44.62	35.84	39.67
NN-CRF-collapsed♮	58.72	45.96	51.56	62.61	60.53	61.56	46.32	32.84	38.36
LSTM-CRF†	58.66	50.92	54.52	67.11	66.45	66.78	52.10	44.62	48.07
HAST-TNet†	57.13	54.62	55.85	63.02	73.94	68.04	46.61	50.02	48.26
UNIFIED♮	61.27	54.89	57.90	68.64	71.01	69.80	53.08	43.56	48.01
BERT-CLS	54.83	61.96	57.59	68.05	71.21	69.59	44.41	52.48	48.11
BERT-CRF-joint	56.43	62.16	59.15	68.29	72.45	70.31	43.95	55.12	48.92
BERT-CRF-collapsed	56.67	62.79	59.57	68.64	72.67	70.59	44.48	55.78	49.48
LCM-Add	59.45	65.25	62.20	68.94	73.50	71.14	47.51	57.54	52.04
LCM-Gate	59.48	65.66	**62.42**	69.06	74.96	**71.89**	48.11	58.74	**52.89**

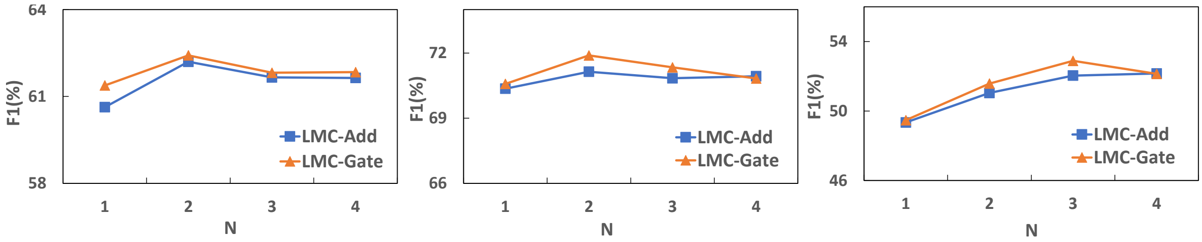
Table 3: Experimental results. The results of the models marked with '†' are reproduced by us with the officially released code by the authors. The results of the models marked with '♮' are copied from Li et al. (2019a). Average results over three runs with random initialization are reported. State-of-the-art results are marked in bold.

model for ABSA task. Third, despite that BERT-$\{$CLS, CRF$\}$ baselines show competitiveness compared with the previous best approach UNIFIED, they are all beaten by our model. For example, LCM-Gate achieves 4.83%, 2.3%, and 4.78% absolute gains on the three datasets compared with BERT-CLS, indicating the effectiveness of our model. Fourth, among the joints methods, LCM-$\{$Add, Gate$\}$ shows the best performance. This illustrates not only the richness of the features extracted by the encoder but also the efficacy of utilizing the phenomenon that aspect boundary label and sentiment label of the same word can correct each other. Last but not least, we notice that the improvement of our model on the Rest dataset is marginal in contrast with the state-of-the-art baseline UNIFIED. The small gap is reasonable since the Rest dataset contains many informal reviews resulting in the inferior modeling power of models. A similar observation is seen in the comparison of UNIFIED and HAST-TNet.

Ablation Study To prove the usefulness of applying one label to correct the other in the joint model, we perform an ablation study. If our model directly employs the hidden states (H^L in Equation 1) to predict the sentiment label sequence and aspect boundary label sequence, our model is equivalent to the BERT-CRF-joint. Consequently, we can choose BERT-CRF-joint as the ablation baseline. As shown in Table 3, compared with LCM-$\{$Add, Gate$\}$, BERT-CRF-joint performs poorly on all three datasets. For example, compared with LCM-Gate, BERT-CRF-joint loses 3.27% F1 scores on the Laptop dataset. This confirms that considering another label embeddings when predicting labels helps improve the performance. Furthermore, we can observe that LCM-Gate is better than LCM-Add on the three datasets. For instance, in terms of F1 scores, LCM-Gate wins about 0.22% absolute gains over LCM-Add on the Laptop dataset. This indicates that the gate mechanism optionally allowing the label embeddings information through is more effective than direct addition.

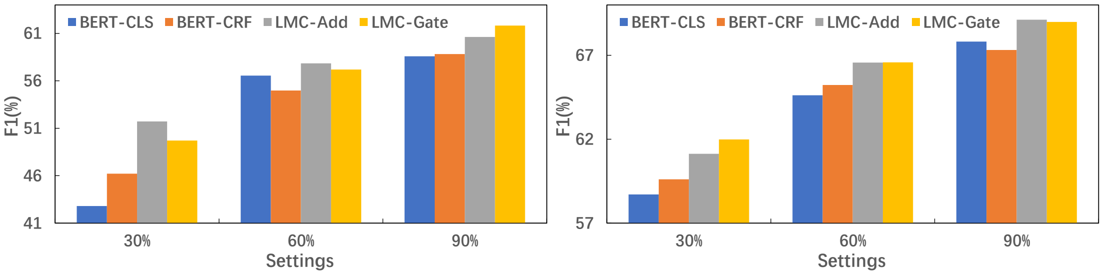
4.4 Discussions

Investigation on the Impact of Number of Repeated Inferences As mentioned above, the inference operation (i.e., rectangular surrounded by the red dotted line in Figure 1) of our model can be repeated multiple times. Here, we investigate the effect of the number of repeated inferences (N) on the performance. We vary the value of N in the set $\{1, 2, 3, 4\}$ and plot the corresponding F1 score of LCM on the three datasets. The results are illustrated in Figure 2. As shown in Figure 2, both models (LCM-Add and LCM-Gate) achieve good performance on Laptop and Rest when N is 2, and on $\mathrm{Twitter}$ when N is 3. This shows that after certain times of inferences, an increase in the times of inferences does not necessarily improve the performance and the model's ability to correct wrong labels may reach a limit. Besides, we discover that increasing N too big may induce the model to overcorrect the label. It is worth noting that although the performance of both models presents a slight rising trend on the $\mathrm{Twitter}$ dataset



(a) Laptop (b) Rest (c) Twitter

Figure 2: Effect of the number of repeated inferences. F1 score on the three datasets is the average value over three runs with random initialization.



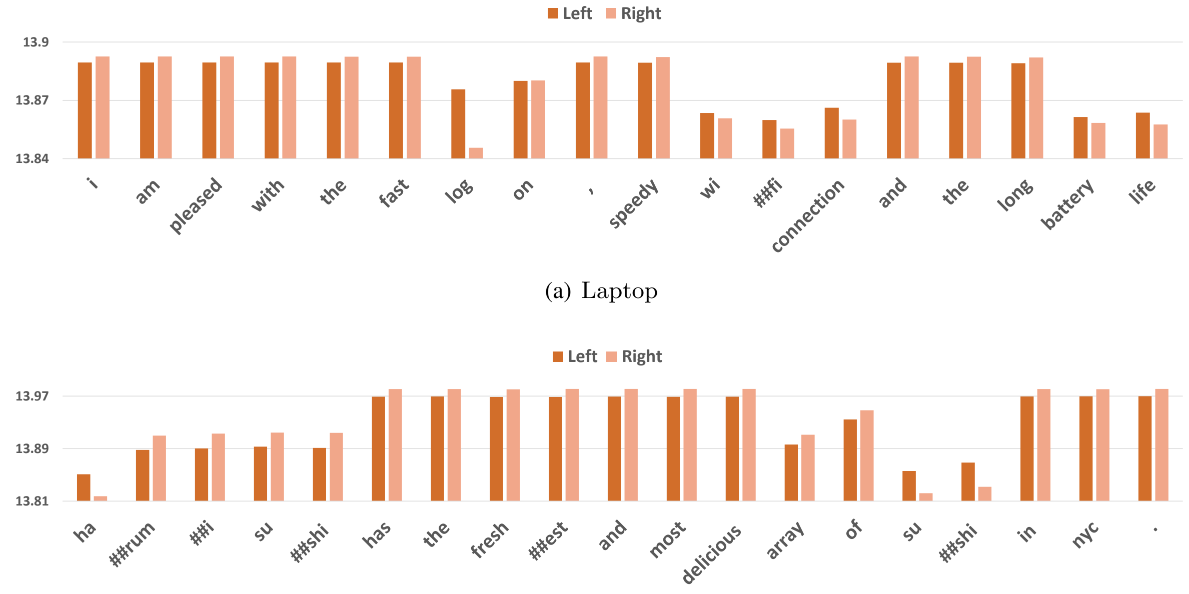
(a) Laptop (b) Rest

Figure 3: Performance against training data size. Here, BERT-CRF refers to BERT-CRF-joint. F1 score on the Laptop and Rest datasets is the average value over three runs with random initialization.

as N increases, we do not take into account it to do trade-off the effectiveness and efficiency.

Performance against Training Data Size. Our motivation is to use the sentiment (aspect boundary) label to correct the aspect boundary (sentiment) label in the joint model. Under certain conditions, the improvement of our model is more significant when the joint model predicts several wrong labels. To this end, we conduct different experiments on different amounts of training data. Figure 3 shows the performance of baselines and our model against different settings for training with different amounts of training data. Here, we consider three training settings (30%, 60%, and 90%) of the original training data. As shown in Figure 3, LCM-{Add, Gate} consistently outperform BERT-{CLS, CRF} under the same amount of training data on the Laptop and Rest datasets, illustrating the superiority of our model. We can observe that the performance gap becomes more obvious when the size of the training data decreases. Concretely, using 30% of the original training data in the Laptop dataset, LCM-Add can achieve an F1 score of 51.72%, higher than BERT-CRF trained on the same size training data. This demonstrates that when the size of the training data is small, the joint model has a larger error space, resulting in more mistaken labels. In contrast, our model using the label embeddings can reduce the error space and correct the mistaken labels.

Gate Mechanism Visualization To confirm that our model can select the valuable sentiment (aspect boundary) label information for predicting the aspect boundary (sentiment) label, we visualize the *2-Norm Gate* value (see Gate Mechanism subsection) in Figure 4. Our observations are as follows. First, we can see that when a token is an aspect word, its *2-Norm Gate* value is relatively small. The underlying reason may be that using only contextual information is not sufficient, thus allowing the label information to pass through the filter. Second, we can find that the different aspect words have different *2-Norm Gate* values (e.g., compare the *2-Norm Gate* values between *log* and *battery* in Figure 4(a)). The reason is that if an aspect word appears more frequently in the dataset, its contextual information can predict correctly its label with the help of little label information. Finally, we observe that the same aspect word has two different *2-Norm Gate* values (e.g., compare Left and Right *2-Norm Gate* values of *log* in Figure 4(a)).



(a) Laptop

(b) Rest

Figure 4: Gate mechanism visualization of an input sentence from the Laptop and Rest dataset. Here, the y-axis is the *2-Norm Gate* value. Left (Right) denotes the *2-Norm Gate* value of the gate mechanism when predicting the sentiment (aspect boundary) label (see Figure 1). In the Laptop sentence, *log on*, *wifi connection*, and *battery life* are aspects, with *positive* as their corresponding sentiments. In the Rest sentence, *array of sushi* is an aspect, with *positive* as its corresponding sentiments.

We assume that it can be inferred correctly using the contextual information when the sentiment or aspect boundary label of an aspect word is easy to predict. For example, *log* is very close to its sentiment cue *pleased*, which results in using the contextual information easy to infer its sentiment label.

Case Study Table 4 presents some qualitative cases sampled from the ablation baseline (i.e., BERT-CRF-joint) and LCM-{Add, Gate}. As observed in the first and second sentences, BERT-CRF-joint correctly predicts the aspect boundary (sentiment) but fails to forecast the right sentiment (aspect boundary). By contrast, LCM, where we incorporate the boundary (sentiment) label embeddings into the hidden states to correct the prediction of the sentiment (boundary) label, can correctly handle these two cases, suggesting that our idea using one label to correct the other is capable of improving the performance of ABSA. Besides, we find that LCM-Add may abandon the correction halfway. For example, in the fourth sentence, it successfully corrects the boundary label of *harumi* with the sentiment label O but

Sentence	**BERT-CRF-joint**		**LCM-Add**		**LCM-Gate**	
	Aspect	Sentiment	Aspect	Sentiment	Aspect	Sentiment
I do not like too much [Windows 8]$_{\text{POS}}$.	Windows 8	POS(✗)	Windows 8	NEG	Windows 8	NEG
Their [duck]$_{\text{POS}}$ here is also absolutely delicious.	duck here(✗)	POS O	duck	POS	duck	POS
The Apple engineers have not yet discovered the [delete key]$_{\text{NEG}}$.	Apple engineers(✗) delete key	NEG(✗) NEG	Apple engineers(✗) delete key	O NEG	delete key	NEG

harumi sushi has the freshest and most delicious [array of sushi]$_{\text{POS}}$ in NYC.	harumi(✗)
	array(✗)
	sushi(✗)
		O
		POS
		POS
			array
			of
			sushi
				POS
				O(✗)
				POS
					array of sushi	POS

Table 4: Case Study. The "Aspect" column contains the results from the aspect boundary label sequence. The "Sentiment" column presents the sentiment of the aspect, coming from the sentiment label sequence. The marker ✗denotes the incorrect prediction.

fails to correct the sentiment label of *of* with the boundary label I. We attribute it to LCM-Add's inability to selectively utilize the label information. In contrast, LCM-Gate performs better and corrects the label more thoroughly, indicating that the gate mechanism, where optionally let the label information through, is more reasonable and effective compared with the pure addition.

5 Conclusions

In this paper, we investigate aspect-based sentiment analysis tasks, which can be formulated as two sequence labeling problems with a set of aspect boundary labels and a set of sentiment labels. We observe that the aspect boundary label and sentiment label of the same word can correct each other. Thus, we propose the label correction model that exploits this observation to improve performance. Moreover, we introduce two ways (add and gate mechanism) to make use of label information. We evaluate our proposed model on the three benchmark datasets. Extensive experiments show that our proposed model achieves superior performance.

Acknowledgements

The authors wish to thank the anonymous reviewers for their helpful comments. This research is partially supported by the National Natural Science Foundation of China (No. U1811462 and U1711263) and Guangdong Province Key Laboratory of Computational Science at the Sun Yat-sen University (2020B1212060032).

References

Peng Chen, Zhongqian Sun, Lidong Bing, and Wei Yang. 2017. Recurrent attention network on memory for aspect sentiment analysis. In *Proceedings of EMNLP*, pages 452–461.

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

G David Forney. 1973. The viterbi algorithm. *Proceedings of IEEE*, 61(3):268–278.

Minqing Hu and Bing Liu. 2004. Mining and summarizing customer reviews. In *Proceedings of KDD*, pages 168–177.

Niklas Jakob and Iryna Gurevych. 2010. Extracting opinion targets in a single-and cross-domain setting with conditional random fields. In *Proceedings of MNLP*, pages 1035–1045.

Long Jiang, Mo Yu, Ming Zhou, Xiaohua Liu, and Tiejun Zhao. 2011. Target-dependent twitter sentiment classification. In *Proceedings of ACL*, pages 151–160.

Wei Jin and Hung Hay Ho. 2009. A novel lexicalized hmm-based learning framework for web opinion mining. In *Proceedings of ICML*, pages 465–472.

John D Lafferty, Andrew McCallum, and Fernando CN Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of ICML*, pages 282–289.

Guillaume Lample, Miguel Ballesteros, Sandeep Subramanian, Kazuya Kawakami, and Chris Dyer. 2016. Neural architectures for named entity recognition. In *Proceedings of NAACL*, pages 260–270.

Xin Li, Lidong Bing, Wai Lam, and Bei Shi. 2018a. Transformation networks for target-oriented sentiment classification. In *Proceedings of ACL*, pages 946–956.

Xin Li, Lidong Bing, Piji Li, Wai Lam, and Zhimou Yang. 2018b. Aspect term extraction with history attention and selective transformation. In *Proceedings of IJCAI*, pages 4194–4200.

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

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

Kang Liu, Heng Li Xu, Yang Liu, and Jun Zhao. 2013. Opinion target extraction using partially-supervised word alignment model. In *IJCAI*.

Pengfei Liu, Shafiq Joty, and Helen Meng. 2015. Fine-grained opinion mining with recurrent neural networks and word embeddings. In *Proceedings of EMNLP*, pages 1433–1443.

Dehong Ma, Sujian Li, Xiaodong Zhang, and Houfeng Wang. 2017. Interactive attention networks for aspect-level sentiment classification. In *Proceedings of IJCAI*, pages 4068–4074.

Margaret Mitchell, Jacqui Aguilar, Theresa Wilson, and Benjamin Van Durme. 2013. Open domain targeted sentiment. In *Proceedings of EMNLP*, pages 1643–1654.

Maria Pontiki, Dimitrios Galanis, John Pavlopoulos, Harris Papageorgiou, Ion Androutsopoulos, and Suresh Manandhar. 2014. Semeval-2014 task 4: Aspect based sentiment analysis. In *Proceedings of SemEval*, pages 27–35.

Maria Pontiki, Dimitrios Galanis, Harris Papageorgiou, Suresh Manandhar, and Ion Androutsopoulos. 2015. Semeval-2015 task 12: Aspect based sentiment analysis. In *Proceedings of SemEval*, pages 486–495.

Maria Pontiki, Dimitrios Galanis, Harris Papageorgiou, Ion Androutsopoulos, Suresh Manandhar, AL-Smadi Mohammad, Mahmoud Al-Ayyoub, Yanyan Zhao, Bing Qin, Orphee De Clercq, et al. 2016. Semeval-2016 task 5: Aspect based sentiment analysis. In *Proceedings of SemEval*, pages 19–30.

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: a simple way to prevent neural networks from overfitting. *JMLP*, 15(1):1929–1958.

Duyu Tang, Bing Qin, Xiaocheng Feng, and Ting Liu. 2016a. Effective lstms for target-dependent sentiment classification. In *Proceedings of COLING*, pages 3298–3307.

Duyu Tang, Bing Qin, and Ting Liu. 2016b. Aspect level sentiment classification with deep memory network. In *Proceedings of EMNLP*, pages 214–224.

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 *Proceedings of NIPS*, pages 5998–6008.

Joachim Wagner, Piyush Arora, Santiago Cortés Vaíllo, Utsab Barman, Dasha Bogdanova, Jennifer Foster, and Lamia Tounsi. 2014. Dcu: Aspect-based polarity classification for semeval task 4. In *Proceedings of SemEval*, pages 223–229.

Qianlong Wang and Jiangtao Ren. 2020. Sequence prediction model for aspect-level sentiment classification. In *Proceedings of ECAI*, pages 2196–2203.

Wenya Wang, Sinno Jialin Pan, Daniel Dahlmeier, and Xiaokui Xiao. 2016. Recursive neural conditional random fields for aspect-based sentiment analysis. In *Proceedings of EMNLP*, pages 616–626.

Wenya Wang, Sinno Jialin Pan, Daniel Dahlmeier, and Xiaokui Xiao. 2017. Coupled multi-layer attentions for co-extraction of aspect and opinion terms. In *AAAI*.

Shuai Wang, Sahisnu Mazumder, Bing Liu, Mianwei Zhou, and Yi Chang. 2018. Target-sensitive memory networks for aspect sentiment classification. In *Proceedings of ACL*, pages 957–967.

Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, et al. 2016. Google's neural machine translation system: Bridging the gap between human and machine translation. *arXiv preprint arXiv:1609.08144*.

Hu Xu, Bing Liu, Lei Shu, and S Yu Philip. 2018. Double embeddings and cnn-based sequence labeling for aspect extraction. In *Proceedings of ACL*, pages 592–598.

Wei Xue and Tao Li. 2018. Aspect based sentiment analysis with gated convolutional networks. In *Proceedings of ACL*, pages 2514–2523.

Meishan Zhang, Yue Zhang, and Duy-Tin Vo. 2015. Neural networks for open domain targeted sentiment. In *Proceedings of EMNLP*, pages 612–621.

Li Zhuang, Feng Jing, and Xiao-Yan Zhu. 2006. Movie review mining and summarization. In *Proceedings of CIKM*, pages 43–50.