IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received February 14, 2020, accepted March 12, 2020, date of publication March 16, 2020, date of current version March 25, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2981139

Interactive Rule Attention Network for Aspect-Level Sentiment Analysis

QIANG LU, ZHENFANG ZHU, DIANYUAN ZHANG, WENQING WU, AND QIANGQIANG GUO
School of Information Science and Electrical Engineering, Shandong Jiaotong University, Shandong 250357, China

Corresponding author: Zhenfang Zhu (zhuzf@sdjtu.edu.cn)

This work was supported in part by the National Social Science Foundation under Award 19BYY076, in part by the Science Foundation of Ministry of Education of China under Award 14YJC860042, and in part by the Shandong Provincial Social Science Planning Project under Award 19BJCJ51, Award 18CXWJ01, Award 18BJYJ04, and Award 17CHLJ07.

ABSTRACT Aspect-level sentiment analysis is a fundamental task in NLP, and it aims to predict the sentiment polarity of each specific aspect term in a given sentence. Recent researches show that the fine-grained sentiment analysis for aspect-level has become a research hotspot. However, previous work did not consider the influence of grammatical rules on aspect-level sentiment analysis. In addition, attention mechanism is too simple to learn attention information from context and target interactively. Therefore, we propose an interactive rule attention network (IRAN) for aspect-level sentiment analysis. IRAN not only designs a grammar rule encoder, which simulates the grammatical functions at the sentence by standardizing the output of adjacent positions, but also constructs an interaction attention network to learn attention information from context and target. Experimental results on SemEval 2014 Dataset and ACL 2014 Twitter Dataset demonstrate IRAN can learn effective features and obtain superior performance over the baseline models.

INDEX TERMS Aspect-level sentiment analysis, grammatical rules, IRAN, interaction attention network.

I. INTRODUCTION

Sentiment analysis, also known as opinion mining [1], [2], is one of the fundamental tasks of natural language processing [3], [4], and it aims to predict the sentiment polarities of the given texts. In recent years, sentiment analysis has successfully extensive applications in catering, e-commerce, hotel and other fields [5]. For example, in the field of catering, sentiment analysis can help customers find the food which they want from a large number of restaurants, and select more popular restaurants based on a large number of user comments [6]. However, traditional sentiment analysis focuses on document-level or sentence-level, it can only analyze the sentiment polarity of the whole document or sentence which only contains opinions about one topic. For both the document-level and the sentence-level sentiment analysis, the decided sentiment polarities are based on the whole document/sentence rather than the topics given in the document/sentence. Obviously, this is not reasonable in many cases. For instance, in the sentence "This restaurant is small but very good environment", for aspect term restaurant, the sentiment polarity is negative, but for aspect term environment, the polarity is positive and therefore the aspect-level sentiment analysis [7] was proposed to address this problem.

Aspect-level sentiment analysis [8] is a subtask of sentiment analysis, and its goal is aspect identification and aspect-level sentiment classification. The aspect-level sentiment analysis aims to predict the sentiment polarities of each specific aspect term in a given sentence. In recent years, attention mechanism [9] has been successfully extensive applications in many natural language processing (NLP) tasks [10], such as text generation [11], [12], machine translation [13], [14] and question answering [15], [16]. Sentiment analysis models for aspect-level have recently been introduced attention mechanism to models and achieved great results [17]–[20]. Both industry and academia have realized the importance of the aspect-level sentiment analysis, and made attempts to model the relationship by designing a series of attention models [21]. However, there are some defects in the traditional aspect-level sentiment analysis models. On the one hand, it uses the conventional LSTM as the extractor of hidden state, and does not consider the influence of the grammar rules [22]

The associate editor coordinating the review of this manuscript and approving it for publication was Bo Shen.

such as sentiment words, negative words and degree words on the classification performance. On the other hand, the attention mechanism is too simple to learn attention information well from context and aspect.

In order to address the above problems, we propose an interactive rule attention network (IRAN) for aspect-level sentiment analysis. Firstly, our model construct four kinds of rule extractors, including general extractor, sentiment extractor, negative extractor and degree extractor, and get the hidden state of the corresponding rules from these extractors. Compared with the traditional aspect-level sentiment analysis models, our model introduces the external knowledge of grammar rules to the model so that can learn more grammar information from hidden states. Secondly, we design an interactive attention mechanism which adopts multi-attention mechanism to learn the mutual information between context and aspect interactively. This not only makes aspect-level attention representation contain context information with grammatical rules, but also makes context attention representation towards aspect contain aspect-level attention representation and general context representation. Experimental results on SemEval 2014 Dataset and ACL 2014 Twitter Dataset demonstrate IRAN can learn effective features and obtain superior performance over the baseline models.

Our main contributions can be summarized as follows:

1) We propose an interactive rule attention network (IRAN) for aspect-level sentiment analysis, which has been proved to be effective to improve the sentiment analysis performance.
2) We constrain grammar rules in the form of regularization and simulate the grammatical functions at the sentence by standardizing the output of adjacent positions.
3) We design an interactive attention mechanism which adopts multi-attention mechanism to learn the mutual information between context and aspect interactively.

The remaining of this paper is organized as follows. After introducing related works in Section 2, we elaborate our proposed methods in Section 3, and then we perform experimental evaluation in Section 4. Finally, we summarize our work and give an outlook of future work in Section 5.

II. RELATED WORK

In recent years, many scholars have been conducted several researches in the traditional field of sentiment analysis. Traditional sentiment classification methods mainly include sentiment classification method based on dictionary and machine learning. Rao *et al.* [23] propose an efficient algorithm and three pruning strategies to automatically build a word-level emotional dictionary for social emotion detection. Tripathy *et al.* [24] apply four different machine learning algorithms such as Naive Bayes [25], Maximum Entropy [26], Stochastic Gradient Descent [27], and Support Vector Machine [28] to classification of human sentiments. The accuracy of different methods is critically examined in order to access their performance on the basis of parameters such as precision, recall, f-measure, and accuracy. Li *et al.* [29] incorporate such prior sentiment information at both word level and document level, in order to investigate the influence each word has on the sentiment label of both target word and context words.

With the development of deep learning, neural network has successfully extensive applications in sentiment analysis model [30]–[33]. Wang *et al.* [34] propose a regional CNN-LSTM model consisting of two parts: regional CNN and LSTM, in order to predict the VA ratings of texts. By combining the regional CNN and LSTM, both local (regional) information within sentences and long-distance dependency across sentences can be considered in the prediction process. Zhou *et al.* [35] propose an attention-based bilingual representation learning model which learns the distributed semantics of the documents in both the source and the target languages, and propose a hierarchical attention mechanism for the bilingual LSTM network.

However, the traditional sentiment analysis model based on neural network can only analyze the sentiment of one topic in the whole document or sentence. To address this problem, Ma *et al.* [36] propose a novel solution to targeted aspect-based sentiment analysis, which tackles the challenges of both aspect-based sentiment analysis and targeted sentiment analysis by exploiting commonsense knowledge. Cheng *et al.* [37] proposes a semi-supervised method for the ATSA problem by using the Variational Autoencoder based on Transformer. The model learns the latent distribution via variational inference. By disentangling the latent representation into the aspect-specific sentiment and the lexical context, the method induces the underlying sentiment prediction for the unlabeled data, which then benefits the ATSA classifier.

With the increasing expansion of Chinese language on the Web, sentiment analysis in Chinese is becoming an increasingly important research field. Peng *et al.* [38] first introduce and summarize the constructions of sentiment corpora and lexica. Then, they conduct a survey of monolingual sentiment classification in Chinese via three different classification frameworks. Finally, they introduce sentiment classification based on the multilingual approach. Recent researches show that the multi-grained aspect target sequence for Chinese sentiment analysis has become a research hotspot. Peng *et al.* [39] study two fusion methods for such granularities in the task of Chinese aspect-level sentiment analysis. They formalize the problem from a different perspective, i.e., that sentiment at aspect target level should be the main focus. Due to the fact that written Chinese is very rich and complex, Chinese aspect targets can be studied at three different levels of granularity: radical, character and word. Yang *et al.* [40] proposes a multi-task learning model for Chinese-oriented aspect-based sentiment analysis, namely LCF-ATEPC. Compared with existing models, this model equips the capability of extracting aspect term and inferring aspect term polarity synchronously. Moreover, this model is effective to analyze both Chinese and English comments simultaneously and the experiment on a multilingual mixed dataset proved its availability.

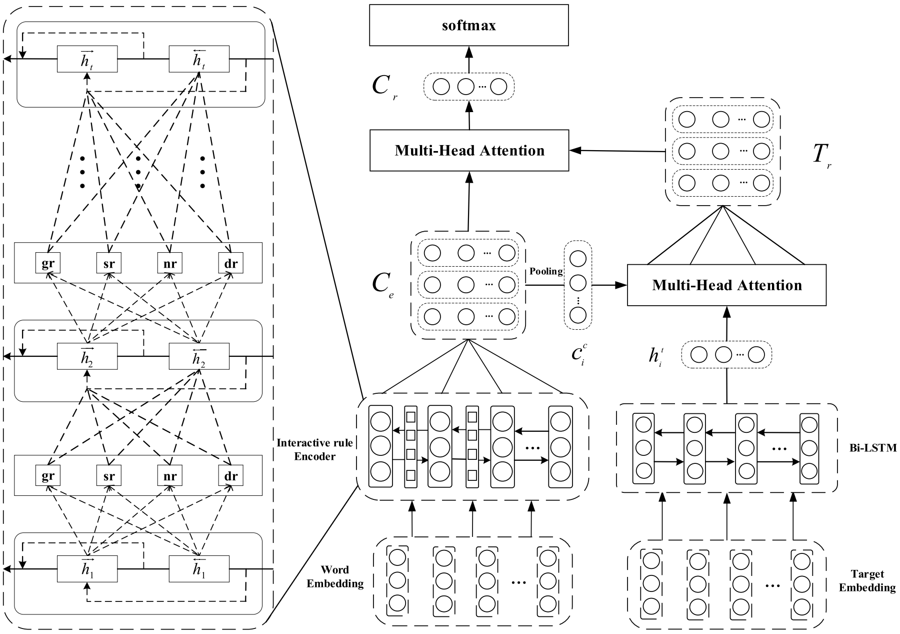


FIGURE 1. The framework architecture of IRAN.

III. MODEL OVERVIEW

In this section, we describe the proposed model Interactive Rule Attention Network (IRAN) for aspect-level sentiment analysis and IRAN is shown in Figure 1. Firstly, we define the research task and notations in the model. Secondly, we introduce an interactive rule encoder to extract the rule information of context. Thirdly, we introduce a new interaction attention network, which could interactively learn attention information from context and target. Finally, we describe the training of the model and loss function with interaction rules.

A. TASK DEFINITION AND NOTATION

The aspect-level sentiment analysis task aims at analyzing the sentiment of sentence towards the target. For example, in the sentence "This restaurant is not big but very good environment", towards "restaurant" is negative, while the sentiment polarity towards "environment" is positive.

Here we introduce some notations to facilitate subsequent descriptions: $X^c = (x_1^c, x_2^c, \cdots, x_n^c)$ is the input sentence, $T^t = (x_1^t, x_2^t, \cdot\cdot\cdot, x_m^t)$ is the given target aspect, $M \in \mathbb{R}^{d\times v}$ is the word embedding matrix, where d denotes the dimension of the embedding, and v indicates the number of words involved in corpus. We input the sentence X^c and the target aspect T^t into the word embedding matrix $M \in \mathbb{R}^{d\times v}$, which can obtain the sentence embedding $E^c = (e_1^c, e_2^c, \cdot\cdot\cdot, e_n^c)$ and aspect embedding $E^t = (e_1^t, e_2^t, \cdot\cdot\cdot, e_m^t)$. We use Bi-LSTM to get the hidden state h_t of common context for each word:

$$\overrightarrow{h_i^c} = LSTM(c_{i-1}^c, \overrightarrow{h_{i-1}^c}, e_i^c)$$
$$\overleftarrow{h_i^c} = LSTM(c_{i+1}^c, \overleftarrow{h_{i+1}^c}, e_i^c)$$
$$h_i^c = [\overrightarrow{h_i^c}; \overleftarrow{h_i^c}] \tag{1}$$

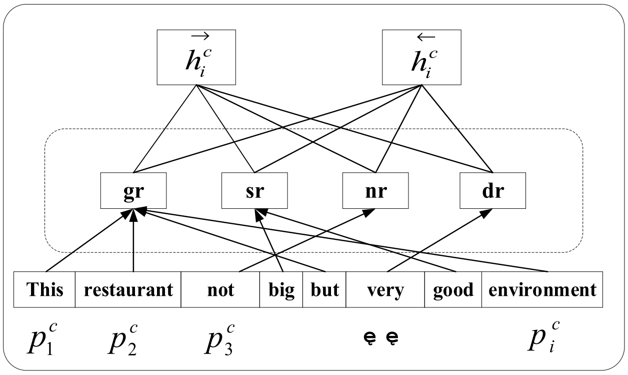


FIGURE 2. The framework architecture of interactive rule encoder.

Similarly, we use Bi-LSTM to get the hidden state of aspects for each word:

$$h_i^t = [\overrightarrow{h_i^t}; \overleftarrow{h_i^t}] \tag{2}$$

B. INTERACTIVE RULE ENCODER

As we all known, the rules of grammar are very important for understanding the context. Take Figure 2 for example, in the sentence "This restaurant is not big but very good environment", the sentiment distributions at "This restaurant" and "restaurant" should be similar by the preprocessing, the sentiment distributions at "big" and "not big" should be opposite, and the sentiment distributions at "good" is enhanced by the degree word "very". On the basis of these rules, we define an interactive rule encoder which include four extractors to simulate this phenomenon. First, we construct four kinds of rule extractors, including general extractor, sentiment extractor, negative extractor and degree extractor, we get the hidden state of the corresponding rules from these extractors. Then, we obtain the hidden representation of rules with the linear change. In the next section, we will explain in detail the construction of four rule extractors and linear variation of hidden state.

In order to simulate the grammatical functions of general, sentiment, negative, degree words, we regularize the difference between the predicted sentiment distribution of the current position, and that of the previous or next positions in model to control these extractors contain the grammatical information. We propose four kinds of rule extractors based on the grammatical phenomenon.

1) GENERAL EXTRACTOR

In the general extractor, if two words in adjacent position are general words, the sentiment distribution of the adjacent positions should be similar. We constrain this rule in the form of regularization and the process can be formulated as follow:

$$R_i^{(gr)} = Relu(W_g \cdot JS(p_i^c||p_{i-1}^c) \cdot h_i^c + b_g) \tag{3}$$

where $R_i^{(gr)}$ indicates the hidden state with general word rules in context, *Relu* is a non-liner activation function, W_g is the weight matrix and b_g is the bias, p_i^c(i.e., h_i^c) is the predicted sentiment distribution at state of position *i*. In particular, $JS(p_i^c||p_{i-1}^c)$ is $Jensen - Shannon\ divergence$ that penalizes the disagreement between p_i^c and p_{i-1}^c. The range of $Jensen - Shannondivergence$ is [0,1], and the probability distributions are the same as 0, but the opposite is 1. According to formula (3), if the adjacent positions are general sentiment words, the sentiment distribution of the two words should be very close and the value of $Jensen - Shannon\ divergence$ should be close to 0.

2) SENTIMENT EXTRACTOR

In the sentiment extractor, if the input word is a sentiment word, the sentiment distribution of the adjacent positions will change, we define this change as sentiment transfer. We construct a sentiment transfer matrix $v \in \mathbb{R}^{v\times d}$, and simulate rules of sentiment transfer by fusing sentiment transfer matrix with hidden state. The sentiment transfer matrix is obtained by model self-training. The process can be formulated as follow:

$$\begin{aligned} p_{i-1}^{(sr)} &= p_{i-1}^c + v_{s(x_i^c)} \\ R_i^{(sr)} &= Relu(W_s \cdot JS(p_i^c||p_{i-1}^{(sr)}) \cdot h_i^c + b_s) \end{aligned} \tag{4}$$

where $p_{i-1}^{(sr)}$ is the sentiment distribution of the hidden state at position *i* after the sentiment transfer, $s(x_i^c)$ is the prior sentiment class of word x_i^c, $R_i^{(sr)}$ indicates the hidden state with sentiment word rules in context. W_s is the weight matrix and b_s is the bias. If the current position is a sentiment word, the sentiment distribution of the adjacent position plusses sentiment transfer distribution will be close to the current position. And according to formula (4), the value of $Jensen - Shannon\ divergence$ should be close to 0. If the sentiment distribution is far away, the value of the $Jensen - Shannon\ divergence$ will increase.

3) NEGATIVE EXTRACTOR

In the negative extractor, if the input word is a negative word, the sentiment polarity of the adjacent positions will be reverse, and the rules of negative words are more complicated than sentiment words. Negative words usually reverse the sentiment polarity in a sentence but sometimes it does not indicate a negative state, for example, the word "good" reflects positive polarity and "bad" reflects negative polarity, the word "not" in "not good" and "not bad" have different rules in polarity change. The former changes the polarity to negative, while the latter changes to neutral instead of positive.

In order to simulate the reverse of negative words, we construct a negation matrix $k \in \mathbb{R}^{d\times v}$. If the input word is a negative word, the sentiment distribution of the current position should be close to that of the next or previous position, which has been reversed. The process can be formulated as follow:

$$\begin{aligned} p_{i-1}^{(nr)} &= p_{i-1}^c \times k_{x_j^c} \\ R_i^{(nr)} &= Relu(W_n \cdot JS(p_i^c||p_{i-1}^{(nr)}) \cdot h_i^c + b_n) \end{aligned} \tag{5}$$

where $p_{i-1}^{(nr)}$ is the sentiment distribution of the hidden state at position *i* after the process of negative matrix. $k_{x_j^c}$ is the negative matrix for a negative word x_j^c, and it is obtained by model self-training. $R_i^{(nr)}$ indicates the hidden state with negative word rules in context. W_n is the weight matrix and b_n is the bias. And according to formula (5), if the current word is a negative word, and the sentiment distribution of adjacent position is similar to that of negative word after the transformation of negative matrix, the value of $Jensen - Shannon\ divergence$ will be closed to 0. If the sentiment distribution is far away, the value of $Jensen - Shannon\ divergence$ will increase.

4) DEGREE EXTRACTOR

In the degree extractor, if the input word is a degree word, the sentiment polarity of the adjacent positions will be enhanced or weaken. For example, the word "very" in "very good" and "very bad" have different rules in polarity change. The former changes the polarity to enhance, while the latter changes to weaken.

In order to simulate the change of degree words, we construct a degree matrix $d \in \mathbb{R}^{d\times v}$. If the input word is a degree word, the sentiment distribution of the current position should be close to that of the next or previous position, which has been enhanced or weakened. The process can be formulated as follow:

$$\begin{aligned} p_{i-1}^{(dr)} &= p_{i-1}^c \times d_{x_l^c} \\ R_i^{(dr)} &= Relu(W_d \cdot JS(p_i^c||p_{i-1}^{(dr)}) \cdot h_i^c + b_d) \end{aligned} \tag{6}$$

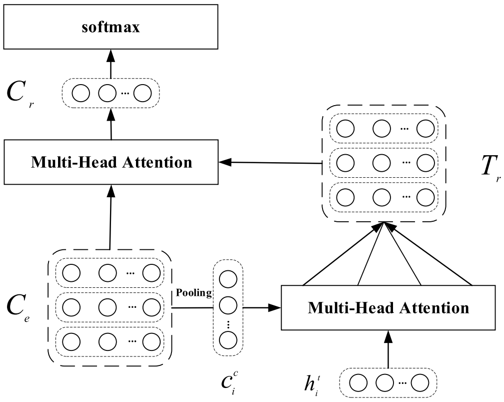


FIGURE 3. The framework architecture of Interactive Attention Network.

where $p_{i-1}^{(dr)}$ is the sentiment distribution of the hidden state at position *i* after the process of degree matrix. $d_{x_l^c}$ is the negative matrix for a negative word x_l^c, and it is obtained by model self-training. $R_i^{(dr)}$ indicates the hidden state with degree word rules in context. W_d is the weight matrix and b_d is the bias. And according to formula (6), if the current word is a degree word, and the sentiment distribution of adjacent position is similar to that of degree word after the transformation of degree matrix, the value of $Jensen - Shannon\ divergence$ will be closed to 0. If the sentiment distribution is far away, the value of $Jensen - Shannon\ divergence$ will increase.

We use these extractors to get the hidden state for rule information and combine it, then make a linear change to form a hidden representation of the context containing the rules. The formulas of building context hidden representation are listed as follows:

$$\begin{aligned} h_{init} &= \sum_{s}^{n} \sum_{i}^{m} R_{i,s} \\ C_e &= W_i h_{init} + b_i \end{aligned} \tag{7}$$

where $R_{i,s}$ is one of these extractors or combination of these extractors on sentence s, h_{init} represents the sum of hidden states for the rules extracted from the context. C_e is the hidden representation of the rules with the linear change. W_i is the weight matrix and b_i is the bias.

C. INTERACTIVE ATTENTION NETWORK

In the previous section, we obtain the hidden representation of rules from the rule extractors. In this section, we propose an interactive attention network which is shown in Figure 3, and the interaction mechanism is designed to learn attention representation interactively between target level and context level attention. We divided this process into two stages which is described in Figure 3.

In the first stage, we first obtain the hidden state of context based on C_e. Then we define an aspect-aware function to get the relationship between each target word and the context representation. Afterwards, we designed a multi-attention mechanism to interactively obtain the normalized attention score of the target and the target attention representation. The process can be formulated as follow:

$$\begin{aligned} T_r &= \sum_{i=1}^{m} \gamma_i \cdot h_i^t \\ \gamma_i &= \frac{exp(f(h_i^t, c_i^c))}{\sum_{j=1}^{m} exp(f(h_j^t, c_i^c))} \\ f(h_i^t, c_i^c) &= relu(W_f \cdot [h_i^t; c_i^c] + b_f) \\ c_i^c &= \frac{1}{n} \sum_{i=1}^{n} C_e \end{aligned} \tag{8}$$

where T_r is the target attention representation, γ_i is the normalized attention score of context towards the target, $f(h_i^t, c_i^c)$ is the aspect-aware function, c_i^c is the hidden state after pooling based on C_e, W_f is the weight matrix and b_f is the bias.

In the second stage, we first get the context attention score based on the target attention representation T_r, then we obtain the finally attention representation of context towards the target. The process can be formulated as follow:

$$\begin{aligned} C_r &= \sum_{i=1}^{n} \alpha_i \cdot h_i^c \\ \alpha_i &= \frac{exp(f(h_i^c, T_r))}{\sum_{j=1}^{n} exp(f(h_j^c, T_r))} \end{aligned} \tag{9}$$

where C_r is the attention representation of context towards the target, α_i is the attention score of the target. Finally, we input the attention representation C_r into the *softmax* layer:

$$\begin{aligned} y &= softmax(W_y x_r + b_y) \\ x_r &= relu(W_x C_r + b_x) \end{aligned} \tag{10}$$

where W_y and W_x are the weight matrix, b_y and b_x are the bias. x_r is the sequence representation and y is the predicted probability distribution.

D. MODEL TRAINING

The purpose of model training is to optimize all the parameters so as to minimize the loss function as much as possible. In order to maintain the accuracy and correlation of the model, we construct a new loss function, which consists of two parts. One part is the rule loss function, and the formulas are as follows:

$$\begin{aligned} L_i &= -\sum_{i}^{N} p_i \log \hat{p}_i + \lambda ||\theta||^2 \\ \hat{p}_i &= softmax(W_i h_{init} + b_i) \end{aligned} \tag{11}$$

where p_i is the correct sentiment polarity, $\hat{p}_i$ is the predicted sentiment polarity for the given sentence, λ is the L_2 regularization parameter and θ is the set of all the parameters in our model. h_{init} represents the sum of hidden states for the rules extracted from the context. W_i is the weight matrix and b_i is the bias.

TABLE 1. Experimental datasets based on SemEval 2014 and ACL 2014 Twitter.

Domain	Positive		Negative		Neutral	
	Train	Test	Train	Test	Train	Test
Restaurant	2160	721	837	184	624	196
Laptop	985	337	860	128	462	169
Twitter	1560	173	1561	173	3127	346

The other part is the original cross entropy loss function, and the construction process of the whole loss function is as follow:

$$loss = -\sum_{i=1}^{N} y_i log(\overset{\wedge}{y_i}) + \mu \sum_{i}^{N} L_i + \frac{1}{2}\lambda||\theta||^2 \qquad (12)$$

Where y_i is the correct sentiment polarity, $\overset{\wedge}{y_i}$ is the predicted sentiment polarity for the given sentence, L_i is the rule loss function. μ is parameter that balances the preference between the cross entropy loss function and the rule loss function. $\frac{1}{2}\lambda||\theta||^2$ is the Regularization term.

IV. EXPERIMENTS

In this section, we will describe the experimental datasets, experimental parameters, experimental comparison models and analysis of IRAN.

A. EXPERIMENTAL DATASETS

We conduct experiments with the datasets of SemEval 2014 task4 and ACL 2014 Twitter to evaluate our model, the SemEval 2014 Dataset consist of reviews in two categories: Restaurant and Laptop, the ACL 2014 Twitter Dataset consist of reviews in one category: Twitter, and the reviews contains three labels of sentiment polarity: {positive, negative, neutral}, the number of each sentiment polarity is shown in Table 1.

B. EXPERIMENTAL PARAMETERS

In our experiments, we use the Glove vector to initialize the word embedding. All the weight matrices get initial values from the uniform distributed U (−0.1, 0.1) and all the biases are set to 0. The learning rate is set to 0.01. The dimension of the word embedding is set to 300, and the number of the hidden units is set to 200. In the training, the number of epoch is set to 30 and the parameter of L2 regularization is set to 10-5.

C. EXPERIMENTAL COMPARISON MODELS

In order to evaluate the performance of our model, we selected several frontier models as baseline models, which include LSTM, AE-LSTM, ATAE-LSTM, PBAN, IAN, Sentic LSTM, BERT, MN, TNET, MN(+ AS) and TNET-ATT(+ AS). We use F1-Measure and Accuracy as evaluation criteria and the experimental comparison results are shown in Table 2.

LSTM [41]: LSTM regard the last hidden vector as the representation of sentence and put the last hidden vector into a softmax layer after linearizing it into a vector, whose length is equal to the number of class labels.

AE-LSTM [41]: AE-LSTM first proposes aspect embedding, and generates attention vector by combining aspect embedding with hidden state.

ATAE-LSTM [41]: ATAE-LSTM is proposed based on AE-LSTM, and append the aspect embedding into each word vector to take better advantage of aspect information.

PBAN [21]: PBAN proposes a method to model the location information of terms in sentences, and use location information to get attention weight of words for final sentiment classification.

IAN [19]: IAN considers the separate modeling of aspect terms and sentences respectively. IAN is able to interactively learn attentions in the contexts and aspect terms, and generates the representations for aspect terms and contexts separately. Finally, it concatenates the aspect term representation and context representation for predicting the sentiment polarity of the aspect terms within its contexts.

Sentic LSTM [36]: Sentic LSTM augment the long short-term memory (LSTM) network with a hierarchical attention mechanism consisting of a target level attention and a sentence-level attention. Commonsense knowledge of sentiment-related concepts is incorporated into the end-to-end training of a deep neural network for sentiment classification.

BERT [42]: BERT uses a bidirectional Transformer network to pre-train a language model on a large corpus, and fine-tunes the pretrained model on other tasks. The first word of the sequence is identified with a unique token [CLS], and a fully-connected layer is connected at the [CLS] position of the last encoder layer, finally a softmax layer completes the sentence or sentence pair classification.

MN [43]: MN proposes a method to solve the problem of target-sensitive sentiment in the above model, and the analysis can be generalized to many existing MNs as long as their improvements are on attention α only.

TNET [44]: TNET proposes a component which consists of L Context-Preserving Transformation (CPT) layers, meanwhile incorporate a mechanism for preserving the original contextual information from the RNN layer.

MN(+ AS) [45]: MN(+AS) is designed based on MN, the context attention weight extracts the correct / wrong prediction for each instance at each iteration, and then mask this word to continue the iteration.

TNET-ATT(+AS) [45]: TNET-ATT(+AS) is designed based on TNET, it uses the same technology as MN to deal with the weight of context attention.

TABLE 2. Comparison with baselines.

Model	Restaurant		Laptop		Twitter	
	Accuracy	F1-Measure	Accuracy	F1-Measure	Accuracy	F1-Measure
LSTM	74.28	-	66.45	-	66.35	61.94
AE-LSTM	76.60	66.45	68.90	62.45	67.70	62.07-
ATAE-LSTM	77.20	65.41	68.70	59.41	-	-
PBAN	81.16	-	74.12	-	-	-
IAN	78.61	70.05	72.16	69.36	72.78	70.02
Sentic LSTM	81.07	70.77	73.69	70.51	-	-
BERT	81.28	71.20	76.54	72.83	73.19	71.69
MN	75.30	64.34	68.90	62.89	67.44	62.37
TNET	80.69	71.27	76.54	71.75	73.02	70.72
MN(+AS)	78.75	69.15	70.53	65.24	71.36	70.17
TNET-ATT(+AS)	81.53	72.90	**77.62**	**73.84**	74.64	73.36
IRAN	**81.96**	**75.02**	74.92	70.29	**74.81**	**73.51**

The results of our model and baseline models on datasets *Restaurant*, *Laptop and Twitter* are shown in Table 2, we find that IRAN has better performance than most frontier models on Accuracy and F1-Measure, which verifies the effectiveness of our model.

LSTM model has the worst performance among all models, because it does not consider the influence of aspect term information on the sentiment polarity of context. The performance of AE-LSTM and ATAE-LSTM is better than LSTM model, because they propose an attention mechanism, which can focus on different parts of a sentence when different aspects are used as input. Specifically, ATAE-LSTM appends the aspect embedding to each word embedding and takes them as inputs on the basis of AE-LSTM, which could get more information related to the terms of the aspect. The performance of IAN is better than ATAE-LSTM, IAN is able to interactively learn attentions in the contexts and aspect terms, and generates the representations for aspect terms and contexts separately. The performance of PBAN model is better than that of IAN and Sentic LSTM. PBAN not only concentrates on the position information of aspect terms, but also mutually models the relation between aspect term and sentence by employing bidirectional attention mechanism. The performance of BERT is slightly better than PBAN. The task-specific BERT design is able to represent either a single sentence or a pair of sentences as a consecutive array of tokens. For a given token, its input representation is constructed by summing its corresponding token, segment, and position embeddings. The performance of MN model is only better than that of LSTM, and MN proposes the target-sensitive memory networks (TMNs) to address the problem which is referred to as target-sensitive sentiment. TNET proposes a component to generate target-specific representations of words in the sentence, meanwhile incorporate a mechanism for preserving the original contextual information from the RNN layer. The performance of TNET is better than that of PBAN on dataset *Laptop*, but worse on dataset *Restaurant*. MN(+AS) and TNET-ATT(+AS) are improved on the basis of MN and TNET, which propose a progressive self-supervised attention learning approach. At each iteration, the context word with the maximum attention weight is extracted as the one with active/misleading influence on the correct/incorrect prediction of every instance, and then the word itself is masked for subsequent iterations. TNET-ATT(+AS) performs better than TNET and achieves an improvement of 0.84 points, 1.08 points and 1.62 points on *Restaurant*, *Laptop* and *Twitter* datasets on Accuracy, and achieves an improvement of 1.63 points, 2.09 points and 2.64 points on F1-Measure. IRAN shows the best performance on *restaurant* and *Twitte* r datasets and achieves an improvement of 0.43 points and 0.17 points than TNET-ATT(+AS) on Accuracy, it also achieves an improvement of 2.12 points and 0.15 points on F1-Measure. But it is worse than that of TNET-ATT(+AS) on *Laptop* dataset, its performance on *Laptop* dataset is 2.70 points lower than that of TNET-ATT(+AS) on Accuracy, and 3.63 points lower on F1-Measure. In IRAN, we combine the grammar rules with model in the form of constraints, and simulate the linguistic functions at the sentence by standardizing the output of adjacent positions, which integrate external knowledge into the

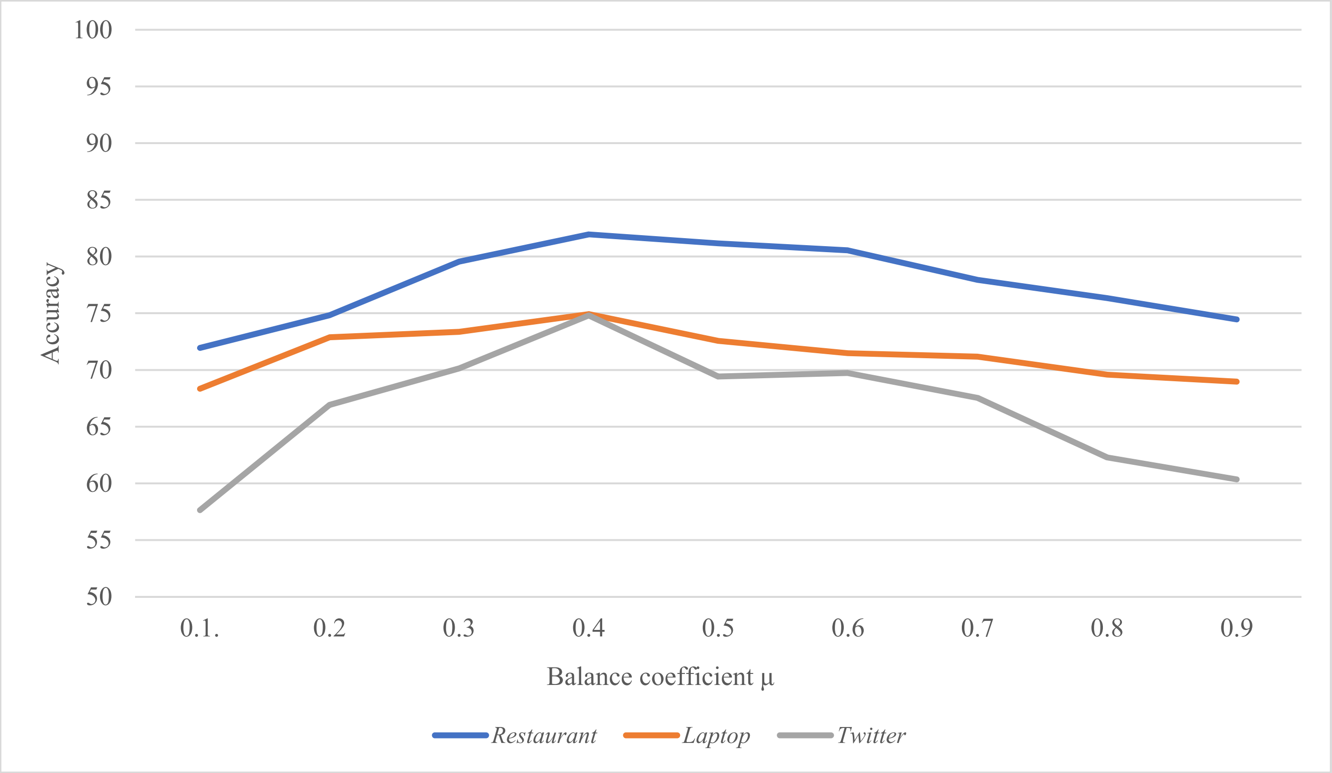


FIGURE 4. The influence of the balance coefficient μ on accuracy.

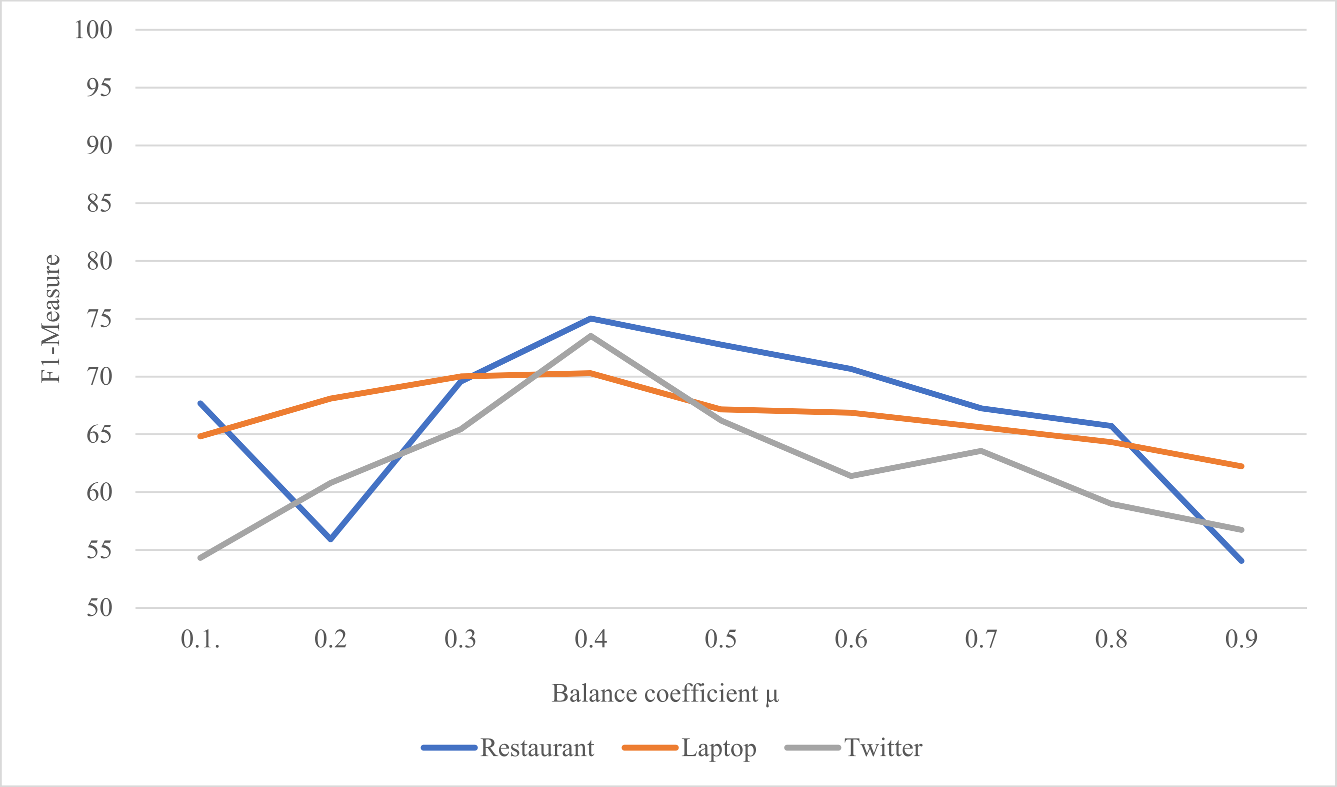
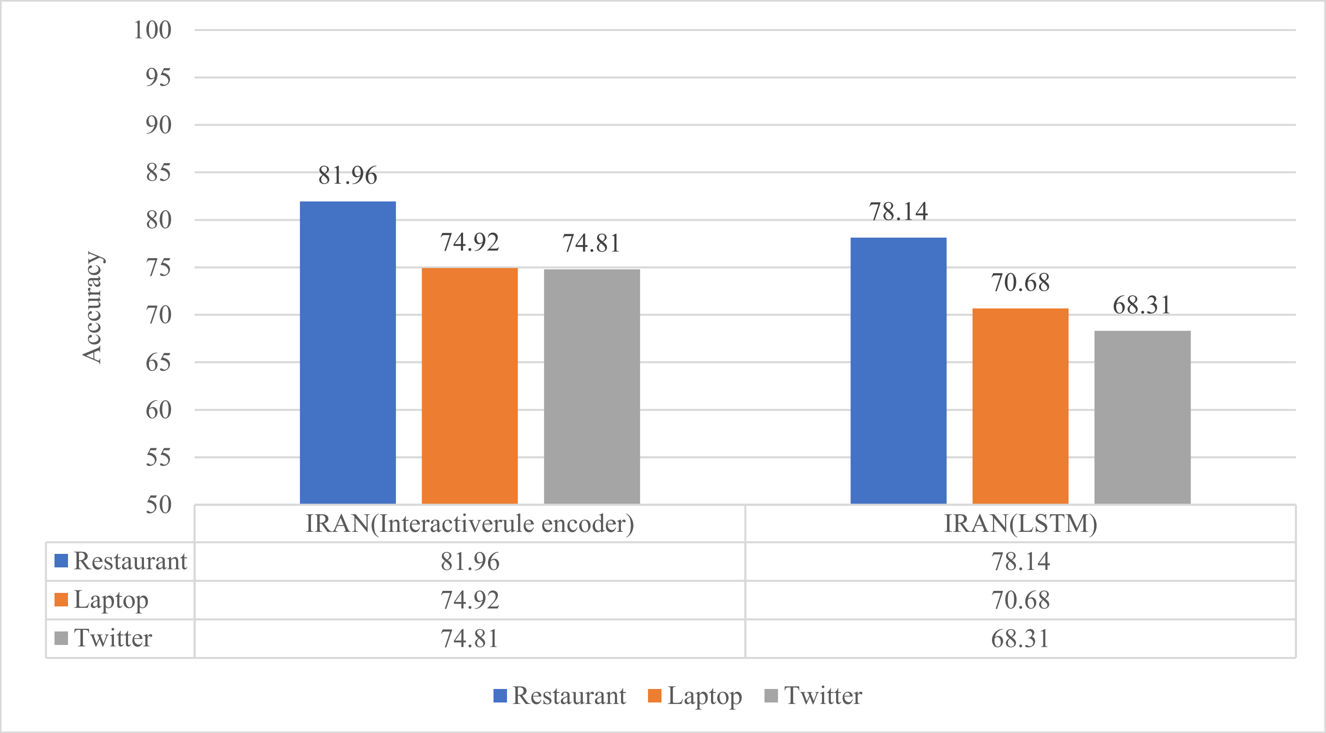


FIGURE 5. The influence of the balance coefficient μ on F1-Measure.



	IRAN(Interactiverule encoder)	IRAN(LSTM)
■ Restaurant	81.96	78.14
■ Laptop	74.92	70.68
■ Twitter	74.81	68.31

FIGURE 6. **The accuracy of the IRAN (interactive rule encoder) and IRAN (LSTM) on three datasets Restaurant, Laptop and Twitter.**

model to improve the performance. Meanwhile, we construct a new interaction attention network, which could interactively learn more information from context and target.

D. ANALYSIS OF IRAN

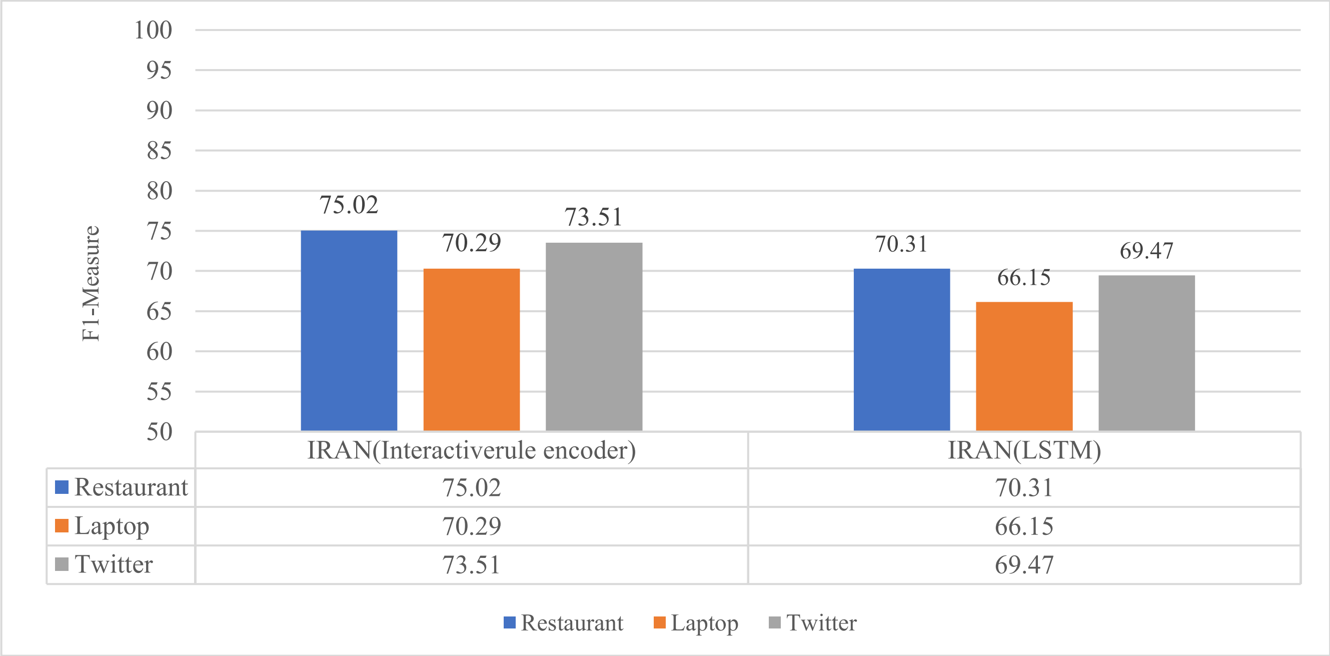
In this section, we design a series of experiments to demonstrate the effectiveness of IRAN. Firstly, we research the influence of balance coefficient μ in formula (12) on the performance. We obtain the hidden state with grammar rule information with rule encoder, then take the hidden state as input to obtain the aspect attention representation which obtains grammar rules, and finally we obtain the context representation with grammar rules toward aspect. The influence of the balance coefficient μ on Accuracy and F1-Measure are shown in Figure 4 and 5.

As shown in Figure 4 and Figure 5. IRAN shows the best performance of Restaurant dataset, Laptop dataset and Twitter dataset on Accuracy and F1-measure when the balance coefficient is 0.4. We find that the curve of accuracy rate shows an upward trend before 0.4, then gets the optimal value at 0.4 and starts to decline after 0.4. The curve of F1-measure value falls first and then rises before 0.4, then obtain the optimal value at 0.4 and starts to decline after 0.4. We combine the grammar rules with model in the form of constraints, it will improve the performance of the model to a certain extent but it does not fully play a part. Verified by the above experiments, the model shows the best performance on Accuracy and F1-measure when the balance coefficient is 0.4. In order to further verify the influence of grammar rules on model, we design six models to demonstrate the effectiveness of grammar rules. The six models are SAN, NAN, DAN, SNAN, SDAN and NDAN. SAN model, whose structure only contains sentiment extractor and others is similar with RAN. NAN and DAN models are similar with SAN, which structure only contain negative extractor and degree extractor. SNAN model is different from SAN, and the difference between these two models is that SNAN contains two extractors which are sentiment extractor and negative extractor. SDAN and NDAN models are similar with SNAN. The experimental results which the balance coefficient is 0.4 are shown in Table 3 and Table 4.

TABLE 3. **The experimental results with different grammar rule extractors on accuracy.**

Dataset	Restaurant	Laptop	Twitter
SAN	80.19	70.85	69.71
NAN	76.16	67.24	61.20
DAN	78.66	68.01	61.95
SNAN	80.98	72.26	71.96
SDAN	81.43	73.35	72.08
NDAN	79.82	69.28	64.36
IRAN	81.96	74.92	74.81

As shown in Table 3 and Table 4, SAN shows the best performance which only contains a single grammar



	IRAN(Interactiverule encoder)	IRAN(LSTM)
■ Restaurant	75.02	70.31
■ Laptop	70.29	66.15
■ Twitter	73.51	69.47

FIGURE 7. The F1-Measure of the IRAN (interactive rule encoder) and IRAN (LSTM) on three datasets Restaurant, Laptop and Twitter.

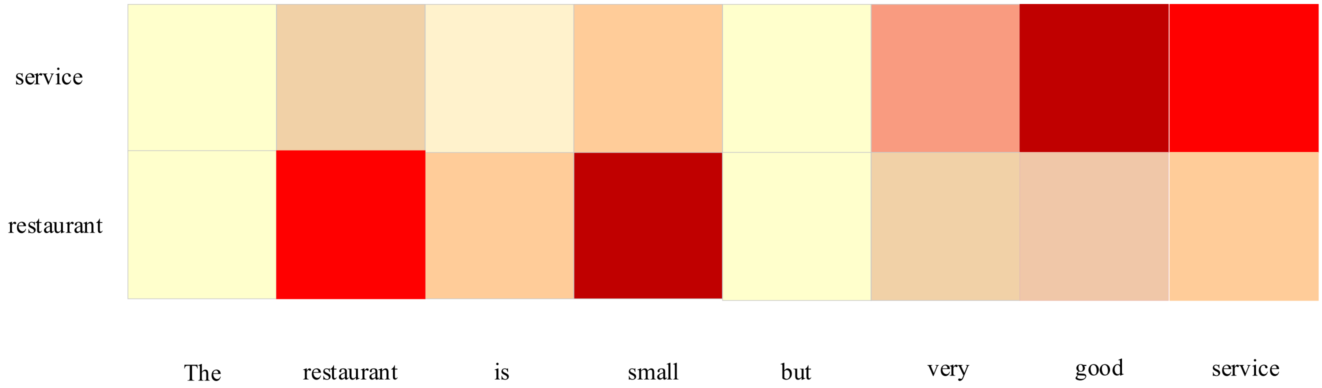


FIGURE 8. Visualized attention weight of sentence and aspect terms.

TABLE 4. The experimental results with different grammar rule extractors on F1-Measure.

Dataset	Restaurant	Laptop	Twitter
SAN	70.79	66.12	63.56
NAN	63.63	62.32	59.17
DAN	68.56	63.89	60.03
SNAN	72.09	67.47	70.74
SDAN	73.98	68.66	71.09
NDAN	71.11	63.51	61.42
IRAN	75.02	70.29	73.51

rule extractor. The main reason is that sentiment words have the greatest impact on the performance of the model. The performance of DAN is slightly higher than that of NAN, and the main reason is that degree words appear more frequently in sentences than negative words. SDAN shows the best performance which contains two extractors. The main reason is that SNAN contains sentiment extractor and degree extractor have the greatest impact on the model. The performance of SNAN is slightly higher than that of NDAN, and main reason is that the sentiment words appear more frequently in sentences than degree words.

To further verify the performance of the grammar rule extractors, we design conventional LSTM instead of the interactive rule encoder, and verify the model by comparing the results of accuracy and F1-Measure on three datasets.

As shown in Figure 6 and Figure 7, the performance of IRAN with interactive rule extractor is better than that with conventional LSTM. This is because interactive rule encoder fully considers the influence of grammar rules in sentences, and simulates the grammatical functions at the sentence by standardizing the output of adjacent positions.

In order to obtain a deeper understanding of IRAN. We visualize the focus of the target words and context words in Figure 8, the darker color means the higher attention.

As shown in Figure 8, the sentence "The restaurant is small but very good service" contains two aspects "restaurant" and "service", but the weight of the context word is different for each aspect word. For example, in terms of the aspect "restaurant", "small" received the highest attention, "good" got a lower level of attention. It effectively avoids the influence of other sentiment words on itself, and pays attention to the sentiment words related to itself. This is because IRAN using an interaction attention network to learn attention information from context and target.

V. CONCLUSION AND FUTURE WORK

In this paper, we propose an Interactive Rule Attention Network (IRAN) for aspect-level sentiment analysis. The main idea of IRAN is to build a grammar rule encoder, which simulate the linguistic functions at the sentence by standardizing the output of adjacent positions. Moreover, IRAN adopts multi-attention mechanism to obtain attention representation between target level and context level attention. Experimental results on SemEval 2014 Dataset and ACL 2014 Twitter Dataset demonstrate that our proposed models can learn effective features and obtain superior performance over the baseline models. In the future work, we will devote to address the defect of the attention mechanism, which is that a few frequent words with sentiment polarities are tend to be overlearned, while those with low frequency often lack sufficient learning.

REFERENCES

[1] B. Liu, "Sentiment analysis and opinion mining," *Synth. Lectures Hum. Lang. Technol.*, vol. 5, no. 1, pp. 1–167, 2012.

[2] B. Pang and L. Lee, "Opinion mining and sentiment analysis," *Found. Trends Inf. Retr.*, vol. 2, nos. 1–2, pp. 1–135, 2008.

[3] E. Cambria, "Affective computing and sentiment analysis," *IEEE Intell. Syst.*, vol. 31, no. 2, pp. 102–107, Mar./Apr. 2016.

[4] E. Cambria, S. Poria, A. Gelbukh, and M. Thelwall, "Sentiment analysis is a big suitcase," *IEEE Intell. Syst.*, vol. 32, no. 6, pp. 74–80, Nov. 2017.

[5] M. Hu and B. Liu, "Mining and summarizing customer reviews," in *Proc. ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining (KDD)*. New York, NY, USA: ACM, 2004, pp. 168–177.

[6] Q. Gan and Y. Yu, "Restaurant rating: Industrial standard and word-of-mouth—A text mining and multi-dimensional sentiment analysis," in *Proc. 48th Hawaii Int. Conf. Syst. Sci.*, Jan. 2015, pp. 1332–1340.

[7] J. Zhou, J. X. Huang, Q. Chen, Q. V. Hu, T. Wang, and L. He, "Deep learning for aspect-level sentiment classification: Survey, vision and challenges," *IEEE Access*, vol. 7, pp. 78454–78483, 2019.

[8] K. Schouten and F. Frasincar, "Survey on aspect-level sentiment analysis," *IEEE Trans. Knowl. Data Eng.*, vol. 28, no. 3, pp. 813–830, Mar. 2016.

[9] P. Chen, Z. Sun, L. Bing, and W. Yang, "Recurrent attention network on memory for aspect sentiment analysis," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2017, pp. 452–461.

[10] P. Zhang, H. Zhu, T. Xiong, and Y. Yang, "Co-attention network and low-rank bilinear pooling for aspect based sentiment analysis," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, May 2019, pp. 6725–6729.

[11] C. Kiddon, L. Zettlemoyer, and Y. Choi, "Globally coherent text generation with neural checklist models," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2016, pp. 329–339.

[12] H. Wang, W. Zhang, Y. Zhu, and Z. Bai, "Data-to-text generation with attention recurrent unit," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, Jul. 2019, pp. 1–8.

[13] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," 2014, *arXiv:1409.0473*. [Online]. Available: http://arxiv.org/abs/1409.0473

[14] T. Luong, H. Pham, and C. D. Manning, "Effective approaches to attention-based neural machine translation," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, Lisbon, Portugal, Sep. 2015, pp. 1412–1421.

[15] A. Kumar, O. Irsoy, P. Ondruska, M. Iyyer, J. Bradbury, I. Gulrajani, V. Zhong, R. Paulus, and R. Socher, "Ask me anything: Dynamic memory networks for natural language processing," in *Proc. Int. Conf. Mach. Learn.*, 2016, pp. 1378–1387.

[16] S. Sukhbaatar, A. Szlam, J. Weston, and R. Fergus, "End-to-end memory networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2015, pp. 2440–2448.

[17] J. Wang, J. Li, S. Li, Y. Kang, M. Zhang, L. Si, and G. Zhou, "Aspect sentiment classification with both word-level and clause-level attention networks," in *Proc. 27th Int. Joint Conf. Artif. Intell.*, Jul. 2018, pp. 4439–4445.

[18] Y. Tay, A. T. Luu, and S. C. Hui, "Learning to attend via word-aspect associative fusion for aspect-based sentiment analysis," in *Proc. AAAI*, 2018, pp. 5956–5963.

[19] D. Ma, S. Li, X. Zhang, and H. Wang, "Interactive attention networks for aspect-level sentiment classification," in *Proc. 26th Int. Joint Conf. Artif. Intell.*, Aug. 2017, pp. 4068–4074.

[20] F. Fan, Y. Feng, and D. Zhao, "Multi-grained attention network for aspect-level sentiment classification," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2018, pp. 3433–3442.

[21] S. Gu, "A position-aware bidirectional attention network for aspect-level sentiment analysis," in *Proc. 27th Int. Conf. Comput. Linguistics.*, 2018, pp. 774–784.

[22] L. Dong, F. Wei, S. Liu, M. Zhou, and K. Xu, "A statistical parsing framework for sentiment classification," *Comput. Linguistics*, vol. 41, no. 2, pp. 293–336, Jun. 2015.

[23] Y. Rao, J. Lei, L. Wenyin, Q. Li, and M. Chen, "Building emotional dictionary for sentiment analysis of online news," *World Wide Web*, vol. 17, no. 4, pp. 723–742, Jul. 2014.

[24] A. Tripathy, A. Agrawal, and S. K. Rath, "Classification of sentiment reviews using n-gram machine learning approach," *Expert Syst. Appl.*, vol. 57, pp. 117–126, Sep. 2016.

[25] C. Troussas, M. Virvou, K. J. Espinosa, K. Llaguno, and J. Caro, "Sentiment analysis of facebook statuses using naive bayes classifier for language learning," in *Proc. IISA*, Jul. 2013, pp. 1–6.

[26] H. Y. Lee and H. Renganathan, "Chinese sentiment analysis using maximum entropy," in *Proc. Workshop Sentiment Analysis Where AI Meets Psychology (SAAIP).*, 2011, pp. 89–93.

[27] T. Günther and L. Furrer, "GU-MLT-LT: Sentiment analysis of short messages using linguistic features and stochastic gradient descent," in *Proc. 7th Int. Workshop Semantic Eval., 2nd Joint Conf. Lexical Comput. Semantics (SEM) (SemEval)*, vol. 2, 2013, pp. 328–332.

[28] G. Gautam and D. Yadav, "Sentiment analysis of Twitter data using machine learning approaches and semantic analysis," in *Proc. 7th Int. Conf. Contemp. Comput. (IC3)*, Aug. 2014, pp. 437–442.

[29] Y. Li, Q. Pan, T. Yang, S. Wang, J. Tang, and E. Cambria, "Learning word representations for sentiment analysis," *Cognit. Comput.*, vol. 9, no. 6, pp. 843–851, Dec. 2017.

[30] M. Zhang, Y. Meishan, Y. Zhang, and D.-T. Vo, "Gated neural networks for targeted sentiment analysis," in *Proc. 30th AAAI Conf. Artif. Intell.*, Mar. 2016, pp. 3087–3093.

[31] S. Chen, C. Peng, L. Cai, and L. Guo, "A deep neural network model for target-based sentiment analysis," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, Jul. 2018, pp. 1–7.

[32] F. Wan, "Sentiment analysis of weibo comments based on deep neural network," in *Proc. Int. Conf. Commun., Inf. Syst. Comput. Eng. (CISCE)*, Jul. 2019, pp. 626–630.

[33] D. Goularas and S. Kamis, "Evaluation of deep learning techniques in sentiment analysis from Twitter data," in *Proc. Int. Conf. Deep Learn. Mach. Learn. Emerg. Appl. (Deep-ML)*, Aug. 2019, pp. 12–17.

[34] J. Wang, L.-C. Yu, K. R. Lai, and X. Zhang, "Dimensional sentiment analysis using a regional CNN-LSTM model," in *Proc. 54th Annu. Meeting Assoc. Comput. Linguistics (Short Papers)*, vol. 2, 2016, pp. 225–230.

[35] X. Zhou, X. Wan, and J. Xiao, "Attention-based LSTM network for cross-lingual sentiment classification," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2016, pp. 247–256.

[36] Y. Ma, H. Peng, and E. Cambria, "Targeted aspect-based sentiment analysis via embedding commonsense knowledge into an attentive LSTM," in *Proc. 32nd AAAI Conf. Artif. Intell.*, 2018, pp. 5876–5883.

[37] X. Cheng, W. Xu, T. Wang, W. Chu, W. Huang, K. Chen, and J. Hu, "Variational semi-supervised aspect-term sentiment analysis via transformer," in *Proc. 23rd Conf. Comput. Natural Lang. Learn. (CoNLL)*, 2019, pp. 961–969.

[38] H. Peng, E. Cambria, and A. Hussain, "A review of sentiment analysis research in chinese language," *Cognit. Comput.*, vol. 9, no. 4, pp. 423–435, Aug. 2017.

[39] H. Peng, Y. Ma, Y. Li, and E. Cambria, "Learning multi-grained aspect target sequence for chinese sentiment analysis," *Knowl.-Based Syst.*, vol. 148, pp. 167–176, May 2018.

[40] H. Yang, B. Zeng, J. Yang, Y. Song, and R. Xu, "A multi-task learning model for chinese-oriented aspect polarity classification and aspect term extraction," 2019, *arXiv:1912.07976*. [Online]. Available: http://arxiv.org/abs/1912.07976

[41] Y. Wang, M. Huang, X. Zhu, and L. Zhao, "Attention-based LSTM for aspect-level sentiment classification," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2016, pp. 606–615.

[42] H. Xu, B. Liu, L. Shu, and S. Y. Philip, "Bert post-training for review reading comprehension and aspect-based sentiment analysis," in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics, Hum. Lang. Technol.*, vol. 1, 2019, pp. 2324–2335.

[43] S. Wang, S. Mazumder, B. Liu, M. Zhou, and Y. Chang, "Target-sensitive memory networks for aspect sentiment classification," in *Proc. 56th Annu. Meeting Assoc. for Comput. Linguistics (Long Papers)*, vol. 1, 2018, pp. 957–967.

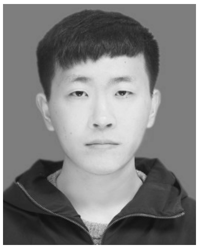
[44] X. Li, L. Bing, W. Lam, and B. Shi, "Transformation networks for target-oriented sentiment classification," 2018, *arXiv:1805.01086*. [Online]. Available: http://arxiv.org/abs/1805.01086

[45] J. Tang, Z. Lu, J. Su, Y. Ge, L. Song, L. Sun, and J. Luo, "Progressive self-supervised attention learning for aspect-level sentiment analysis," in *Proc. 57th Annu. Meeting Assoc. Comput. Linguistics*, 2019, pp. 1–10.

QIANG LU received the B.E. degree in computer science and technology from Shandong Jiaotong University, Jinan, China, in 2016, where he is currently pursuing the M.E. degree. His research interests include sentiment analysis, text classification, and deep learning.

ZHENFANG ZHU received the Ph.D. degree from Shandong Normal University, in 2012. He was a Postdoctoral Fellow with Shandong University, from 2012 to 2015. He is currently a Professor and a Master's Supervisor with the School of Information Science and Electrical Engineering, Shandong Jiaotong University, China. His research interests include network information security, natural language processing, and applied linguistics.

DIANYUAN ZHANG received the B.E. degree in computer science and technology from Shandong Jiaotong University, Jinan, China, in 2019, where he is currently pursuing the M.E. degree. His research interests include sentiment analysis, text classification, and deep learning.



WENQING WU received the B.E. degree in computer science and technology from Shandong Jiaotong University, Jinan, China, in 2019, where he is currently pursuing the M.E. degree. His research interests include data mining, automatic question and answer, and deep learning.

QIANGQIANG GUO received the B.E. degree in electronic information science and technology from Taishan University, Taian, China, in 2018. He is currently pursuing the M.E. degree with Shandong Jiaotong University. His research interests include natural language processing, question and answer systems, and deep learning.