A Self-Attentive Convolutional Neural Networks for Emotion Classification on User-Generated Contents

YING QIAN, WEIWEI LIU, AND JIANGPING HUANG

School of Software Engineering, Chongqing University of Posts and Telecommunications, Chongqing 400065, China

Corresponding authors: Ying Qian (ying05cq@gmail.com) and Jiangping Huang (huangjp@cqupt.edu.cn)

This work was supported in part by the Doctoral Program of Chongqing Federation of Social Science Circles under Grant 2018BS24, and in part by the Humanities and Social Sciences Research Program of Chongqing Education Committee under Grant 19SKGH046.

ABSTRACT

It is challenging to detect emotions on user-generated contents (UGC) because it tends to have sparse emotional semantics, multiple emotions in the same text, and a fast update of emotional expression. Word embedding can extract high-level features in words to enrich their semantics. The convolutional neural networks (CNNs) can make model training more efficient to adapt to fast-changing emotional expressions, but its original pooling operation cannot simultaneously filter out multiple emotional features that are beneficial to classification results, which is not suitable for UGC. In this paper, we propose a self-attentive convolutional neural networks (SACNNs) trained on top of pre-trained word vectors for emotion detection on UGC, which reserves different kinds of emotion information, avoids the loss of emotional aspects in the pooling process of CNNs structure, and increases the interpretability of the model by visualizing the extraction process of features. Meanwhile, the convergence speed of the model training is accelerated, and the model is updated in real time to detect emotions with rich and novel expressions. The proposed model combines the CNNs and the self-attention mechanism, the self-attention mechanism can select key emotional features after convolution. We evaluate the proposed model with two datasets from NLPCC 2014 and SemEval 2018 Task 1. The experimental results show that our model obtain significant performance than baselines in multi-label classification on UGC. In addition, the experimental results and the rationality of the self-attention mechanism are analyzed in detail, and the influential convolutional filter windows are visualized based on attention weights.

INDEX TERMS

Convolutional neural networks, emotion recognition, self-attention mechanism, user-generated contents.

I. INTRODUCTION

Social media such as Twitter, Sina Weibo provide an open platform for people to share their ideas or opinions and generates masses of personalized texts. Most of these texts also called user generated contents (UGC), which are subjective and contain rich emotional information. According to the research of emotions classification on UGC, we can predict the possible behavioral tendencies, preferences and mental states of human beings, which has broad application prospects, such as human-computer interaction dialogue system [1], product recommendation [2], [3] and so on.

Existing emotion classification methods mainly based on dictionary and feature representation. The dictionary-based method estimates the emotions by hand-crafted features, such as sentiment lexicons, which produce a high efficiency on experimental results, but the accuracy depends on the input words, and updating the dictionary is a challenge. The feature-based method mainly adopts machine learning methods. By selecting appropriate models and features to classify text, it has higher classification accuracy, but the efficiency is generally lower than that based on emotion dictionary.

However, UGC published on social media tends to be semantically sparse due to lack of emotional words, and the way it expresses emotions changes quickly. In addition, the free and casual atmosphere in social media often leads to a variety of novel emotion expressions, which greatly increases difficulty of the emotion classification. As can be seen from FIGURE 1, UGC have fewer emotional words, less semantic information, and multiple different emotions in one sentence, which greatly hinders the method based on emotional lexicon. Therefore, a method that avoids depending on the emotional
vocabulary, and can extract the emotional semantic features from the given text in a self-disciplined way is needed. According to these characteristics, the feature-based method is more suitable for UGC. However, the traditional statistical supervised learning methods still depend on the manual features [4], which is hard to express deep emotional semantic features of words.

With the development of deep learning, the word embedding has surpassed the traditional method to represent words in performance [5], alleviates the sparse semantics problem of UGC, and can learn high-level features automatically. Typical deep learning models such as CNNs [6], [7], RNNs [8] and transformer [9] have achieved remarkable results in the fields of NLP. Although transformer has powerful feature extraction capability, its converges slowly while training, which does not adapt to the immediate changes of emotion expression in data corpus. RNNs can learn long-term dependence in sequence data [8], but it can only contain limit sequence structure information. For UGC with less words and sparse semantics, compared to RNNs, it is an advisable choice to choose the CNNs to classify emotions as a more efficient and resource-intensive way. In addition, the existing works have shown that CNNs is as effective as RNNs in many basic NLP tasks [10]. CNNs extracts local semantic features of text through convolutional operations, then filters feature by pooling operation, which is flexible and efficient. There are three common pooling operations, including max pooling, which preserves the largest value of the convolutional layer; $k$-max pooling, which takes top $k$ features and retains their relative positions; and chunk pooling, which divides the feature vector into chunks, and each chunk takes the largest eigenvalue. The three typical pooling operations are shown in FIGURE 2. As can be seen from FIGURE 2, max pooling cannot retain sequence information and intensity feature, and will lose important information [11]. While $k$-max pooling and chunk pooling can retain the relative position information of the feature and capture the intensity feature to some extent, but these methods still cause loss of feature information, especially in the multi-label classification, they are not purposeful to screen out features associated with different classification results, and still cannot capture accurate sequence information and intensity feature. In response to this, we hope to extract the most relevant emotional features of text while retaining the both sequence and intensity information.

In this paper, we propose the self-attentive convolutional neural networks (SACNNs) trained on top of pre-trained word vectors to classify multiple emotions on UGC. Three kinds of convolutional filters with large differences in size are applied to extract the semantic features of word combinations, phrases, and short sentences. In order to connect and encode the three different dimensional features into a fixed-size representation, the self-attention mechanism is used to
replace the original pooling layer in the CNNs, which associates the attention weight with the sequence when extensively extracts key elements by attention weights, and avoids missing other important features. In addition, we visualize the key convolutional filter windows selected by the self-attention mechanism and do a qualitative analysis to verify the rationality of our model.

The main contributions of this paper are as follows:
- We propose SACNNs to effectively extract multiple emotional semantic features on UGC and accelerate the convergence of the model. In addition, the architecture of our model has strong scalability.
- Our method can intuitively visualize the process of extracting key emotional semantics from convolutional filter windows based on attention weights, which increases the interpretability of the model.
- Our model has strong robustness and generalization ability, and can be applied to UGC of different domains, which outperforms most existing text classification models in accuracy and efficiency.

II. RELATED WORK
The feature representation of sentences plays a crucial role in text classification. In this section, some literatures of the current research on sentence representation for text classification and the self-attention mechanism are reviewed.

A. SENTENCE REPRESENTATION FOR TEXT CLASSIFICATION
Text classification has evolved from dictionary-based methods to statistical machine learning methods to deep learning methods [12]. The representation of sentences is the basis of text classification, we need to obtain the word form, syntax and semantic information of the text first, and then classify the text according to the features.

Most of existing works on text classification are based on sentence representation [13], which plays an important role [14]. Existing representation models can be divided into four types: bag-of-words model, sequence representation, structural representation, and attention-based representation [15]. The bag-of-words model [16] uses the index of the word to represent the sentence, but cannot represent the order and syntactic structure information. Joulin et al. [17] proposed word bags combined with n-gram bags to characterize sentences and share information between categories through hidden representations, which is much more efficient than deep learning. The sequence representation model(CNNs) [6], [7] can represent the word order, but cannot represent the structural information of the text. In addition, Gao et al. [18] proposed convolutional filters combined with self-attention mechanism into hierarchical structures to create a document classification model. Using the tags on the document and its constituent sentences, Zhang et al. [19] proposed a sentence-level convolution model estimated the probability that a given sentence affects the classification result and scaled it into the total document representation for classification. Li et al. [20] proposed a model encoded convolutional filters based on semantic features, so that the model can effectively learn features during the training process. These methods have achieved good results in text classification, but it is difficult to avoid the loss of some important features in the pooling layer and identify different emotional semantics.

The structure representation model(RNNs) [8] can represent the text structure and learn the context information, but it also means that the long-term information needs to be ordered into other processing units before entering the current processing unit. The propagation process is complicated and easily leads to gradients disappear. Although the Long Short-Time Memory (LSTM) [21], the Gated Recurrent Unit(GRU) [22] can alleviate the problem of gradient disappearance to some extent, but their memory of long-term information is limited. Zhang et al. [23] proposed a CNNs structure to extract features from the hidden state of the LSTM layer, which can complete the overall structure over a long period of time. Wang et al. [24] proposed a rich mixed semantics structure to improve Chinese sentence representation by using the semantic information of the internal structure of Chinese words. These models have improved the performance of the structural representation model in different aspects, but the slow training efficiency of RNNs cannot adapt to the rapid changes of emotion expression in UGC.

The attention-based model [25] uses the attention mechanism to construct representations by scoring the input words. The attention mechanism originated in the field of image vision, and Bahdanau et al. [26] applied the attention mechanism to the NLP field at the first time. Since then, a large number of variants have been derived based on the attention mechanism. Hu and Dichao [27] has divided the variants of the attention mechanism into seven categories. These different types of attention mechanisms have achieved good results in efficiency and performance. Li et al. [28] proposed the hierarchical attention mechanism, which summarized all past coding vectors into context vectors, which greatly expanded the maximum length of the learning sequence that the RNNs architecture can be learnt. Lin et al. [29] proposed a self-attention mechanism that computed the interactive representation of a sequence by associating different positions of a single sequence. Shin et al. [30] integrated word embedding and attention mechanisms into CNNs for sentiment classification. The attention mechanism is more efficient in filtering the key semantic information with its low computational complexity, which is one of the indispensable technologies in the future.

B. SELF-ATTENTION MECHANISM
Self-attention mechanism is a special kind of attention mechanism. It is characterized by the direct calculation of the dependence between the distances of the words. It can learn the internal structure of sentences and be calculated in parallel, which also has low computational complexity.
The definition of the self-attention mechanism is as follows [29]

$$A = \text{softmax}(W_2 \tanh(W_1 H^T))$$

(1)

where \(W_1\) is a weight matrix, \(W_2\) is a weight matrix that represent the different parts we want to extract from the sentence, and \(H\) is the feature representation of the model.

Self-attention can be used as a layer with RNNs, CNNs, which successfully applied to other NLP tasks. Tan et al. [31] proposed a model with self-attention mechanism, and applied it to the Semantic Role Labeling (SRL) task, which achieved advanced results.

III. SELF-ATTENTIVE CONVOLUTIONAL NEURAL NETWORKS

This section proposes the SACNNs model, shown in FIGURE 3, is a slight variant of the CNNs architecture of [32]. The self-attention mechanism is used to select the important features after the convolutional layer and concatenate them to get the fixed dimension features matrix. Then we put this matrix into the non-linear hidden layer, and finally the output layer makes emotion multi-label prediction.

**A. CONVOLUTIONAL LAYER**

Let \(x_i \in \mathbb{R}^k\) be the \(k\)-dimensional word vector corresponding to the \(i\)-th word in the sentence. A sentence of length \(n\) (padded where necessary) is represented as

$$X_{1:n} = x_1 \oplus x_2 \oplus \ldots \oplus x_n$$

(2)

where \(\oplus\) is the concatenation operator. In general, let \(x_{i:j}\) refer to the concatenation of words \(x_i, x_{i+1}, \ldots, x_j\). Suppose there are \(h\) kinds of convolutional filter window sizes, then the size convolution windows can be expressed as

$$Q = [q_1, q_2, \ldots, q_h]$$

(3)

where \(q_i\) is the size of convolutional filter windows. We apply \(e\) filters for each convolutional filter window of different sizes to produce features. For example, a feature \(c_i\) in the sentence \(X_{1:n}\) is generated from a window of size \(q_i\) by

$$c_i = f(w_c \cdot x_{i:i+q_i-1} + b_1)$$

(4)

where \(b_1 \in \mathbb{R}\) is a bias term and \(f(\cdot)\) is a non-linear function such as the hyperbolic tangent. Each filter with size \(q_i\) is applied to each possible convolutional filter window of words in the sentence with length \(n\) to produce a feature

$$c_i = [c_{i:q_i}, c_{i+q_i+1}, \ldots, c_{n-q_i+1:n}]$$

(5)

with \(c_i \in \mathbb{R}^{n-q_i+1}\), and the dimensions of features extracted by filter with same size \(q_i\) is \(e\)-by-\((n-q_i+1)\). Thus, each possible \(e\) filter of size \(q_i\) windows applied to sentence ultimately produces a total feature map set

$$C = [c_1, c_2, \ldots, c_h]$$

(6)

where the number of features is the kinds of convolutional filter windows size \(h\).

**B. SELF-ATTENTIVE LAYER**

As there are \(h\) convolutional filter window sizes to produce features with unfixed dimensions, the self-attention mechanism is used to encode a variable length feature into a fixed size embedding. The self-attention mechanism takes each feature \(c_i\) extracted from same convolutional filter windows size \(q_i\) as input, and outputs a vector of weights \(A_i\):

$$A_i = \text{softmax}(W_2 \tanh(W_1 c_i^T))$$

(7)

where \(W_1\) is a weight matrix with a shape of \(d_a\)-by-\((n-q_i+1)\). \(W_2\) is a weight matrix with a shape of \(d_a\)-by-\(r\), where \(d_a\) is a hyperparameter we can set arbitrarily, \(r\) is the parts we want to extract from the sentence. Since \(c_i\) is sized \(e\)-by-\((n-q_i+1)\), the annotation vector \(A_i\), have a size \(e\)-by-\(r\). The \(\text{softmax}(\cdot)\) ensures all the computed weights sum up to 1.

Then we sum the feature map set \(c_i\) through the convolutional layer according to the weight provided by \(A_i\), and get the feature embedding vector \(M_i\):

$$M_i = A_i c_i$$

(8)

where \(M_i\) is a feature representation with a shape of \(r\)-by-(\(n-q_i+1\)). It focuses on the key components of the convolutional filter extraction feature, such as a special set of related words, phrases or phrases containing important statistical information. The feature representation after the convolutional layer is obtained by convolutional filter windows with different size. The output of these different filters contains each local semantic feature in the text, which together form the overall semantics of the entire sentence. Therefore, in order to represent overall semantics of the sentence, the \(r\)-dimensional feature representation is chosen to extract different parts from the sentence.

Finally, nonlinear activation operation is performed on the feature embedding vector to obtain a new vector representation \(M'_i\):

$$M'_i = f(W_3 \cdot M_i + b_2)$$

(9)

here \(M'_i\) is a vector with fixed dimension \(r\)-by-(\(n-q_i+1\))-by-\(g\), where \(g\) is a hyperparameter we can set arbitrarily, \(b_2\) is a bias term and \(f(\cdot)\) is a non-linear function.

**C. CONNECTION LAYER**

From above, each size of convolutional filter windows finally can be represented by \(M'_i\), so we connect feature \(M'_i\) extracted by all \(h\) kinds of convolutional filter window size together and get the global feature representation \(M\):  

$$M = M'_1 + M'_2 + \ldots + M'_h$$

(10)

where the size of \(M\) is \(h\)-by-\(g\). Finally the probability distribution over all the classes \(O\) can be obtained by

$$O = f(W_4 \cdot M + b_3)$$

(11)
where $W_4$ is a weight matrix with a shape of $g$-by-$z$, $z$ represents the num of classes for classification tasks, and $b_3$ is a bias term. The softmax operation over the scores of all the classes is calculated as follows:

$$P = \frac{1}{1 + e^{Q}}$$  \hspace{1cm} (12)

We take cross entropy as the loss function that measures the discrepancy between the real emotion distribution $P_t$ and the model output distribution $P$ of sentences in the corpora.

$$loss = P_t \ast (-\log(P)) + (1 - P_t) \ast (-\log(1 - P))$$  \hspace{1cm} (13)

The entire model is trained end-to-end with stochastic gradient descent.

IV. EXPERIMENTS

In this section, we first introduce the data sets, evaluation, and typical multi-label classification models as baselines. Then, the experimental results of this model are compared with baselines. In addition, the data concentration noise is used as a single variable to verify the robustness of the model and analyze the model performance. Finally, according to the self-attention mechanism, the semantic analysis of the text extracted from the convolution is visually analyzed (the heat map of the sentence structure in convolutional process). The feasibility of our model is proved.

A. DATASETS

To verify the validity of our model for emotional classification on UGC, we use two datasets to evaluate the proposed model in the experiment.

1) NLPCC 2014

The Natural Language Processing and China Computing (NLPCC) conference is the annual meeting of the CCF TCCI (China Computer Society China Information Technology Committee). The dataset of task: Chinese Weibo Text Emotion Analysis is selected. The largest Chinese weibo site - Sina Weibo directly provides 20,000 weibos, of which 10,194 weibos contain no emotions, 9806 weibos contain emotions, and their target emotion categories include none, anger, disgust, fear, happiness, love, sadness and surprise. And the task is to mark two emotions for each weibo text.
TABLE 1. Emotion tag in tweets.

| Emotion Tag | Include Emotion               |
|-------------|-------------------------------|
| anger       | annoyance, rage               |
| anticipation| interest, vigilance           |
| disgust     | disinterest, dislike and loathing |
| fear        | apprehension, anxiety, concern, and terror |
| joy         | serenity, ecstasy             |
| love        | affection                     |
| optimism    | hopefulness, confidence       |
| pessimism   | cynicism, lack of confidence  |
| sadness     | pensiveness, grief            |
| surprise    | distraction, amazement        |
| trust       | disability, liking, and admiration |

FIGURE 4. Statistics on the sequence length of the dataset.

2) SEMEVAL 2018
This is the subtask E: Detecting Emotions (multi-label classification) of Task 1: Affect in Tweets in International Workshop on Semantic Evaluation in 2018, and each tweet has 11 emotional tags, as shown in the TABLE 1.

The emotion of each tweet is marked with a binary tag. In addition, the dataset’s emotional classification is less balanced. There are 10,983 tweets, including 10,690 tweets containing emotions and 293 tweets without emotions.

In addition, we count the sequence length of the text in the above two datasets. As shown in the FIGURE 4, the length of tweet usually does not exceed 50, most of weibo text sequence length is less than 120.

B. EVALUATION
In the multi-label emotion classification task, average precision(Ap) of document-pivoted rankings, micro-averaged F scores(Micro-F) and macro-averaged F scores(Macro-F) are chosen to evaluate the performance of our model:

Average precision:

$$A = \frac{1}{n} \sum_{i=1}^{n} \frac{1}{|Y_i|} \sum_{l=1}^{|Y_i|} \frac{|\{E_k \in Y_l | R(x_i, emo_k) \leq R(x_i, y)\}|}{R(x_i, emo_k)}$$

(14)

where R(x_i, y) is the position of emotion y label in document x_i. E_k are the emotions of document. |Y_i| is the number of emotions in document x_i. This measure evaluates the average fraction of labels ranked above a particular label l ∈ Y_i, which actually are in |Y_i| [33].

TABLE 2. Performance of the models on the test dataset of NLPCC 2014 task: Chinese Weibo Text Emotion Analysis.

| Models       | With noise | Without noise |
|--------------|------------|---------------|
| fastText     | 0.616      | 0.856         | 0.6820.644 | 0.819 | 0.617 |
| HINA         | 0.536      | 0.836         | 0.639 | 0.463 | 0.770 | 0.555 |
| RNNs         | 0.640      | 0.841         | 0.651 | 0.563 | 0.804 | 0.626 |
| RCNNs        | 0.504      | 0.744         | 0.491 | 0.451 | 0.746 | 0.510 |
| SACNN        | 0.556      | 0.788         | 0.615 | 0.594 | 0.813 | 0.625 |
| CNN-static   | 0.620      | 0.859         | 0.686 | 0.629 | 0.826 | 0.628 |
| CNN-non-static| 0.671    | 0.864         | 0.690 | 0.632 | 0.841 | 0.654 |

TABLE 3. Performance of the models on the test dataset of semeval 2018 subtask E: Detecting Emotions (multi-label classification) of task 1: Affect in Tweets.

| Models       | With noise | Without noise |
|--------------|------------|---------------|
| fastText     | 0.648      | 0.686         | 0.497 | 0.483 | 0.697 | 0.563 |
| HINA         | 0.657      | 0.820         | 0.707 | 0.653 | 0.788 | 0.683 |
| RNNs         | 0.511      | 0.759         | 0.603 | 0.500 | 0.727 | 0.592 |
| RCNNs        | 0.458      | 0.740         | 0.575 | 0.413 | 0.679 | 0.525 |
| SACNN        | 0.470      | 0.741         | 0.571 | 0.538 | 0.742 | 0.610 |
| CNN-static   | 0.733      | 0.855         | 0.767 | 0.602 | 0.768 | 0.657 |
| CNN-non-static| 0.739    | 0.870         | 0.776 | 0.651 | 0.812 | 0.695 |
| SACNN-static | 0.741      | 0.865         | 0.769 | 0.675 | 0.874 | 0.823 |
| SACNN-non-static| 0.806   | 0.883         | 0.812 | 0.684 | 0.882 | 0.833 |

Micro-averaged F-score:

$$P_1 = \frac{\sum_{e \in E} N_{correct \rightarrow emo}}{\sum_{e \in E} N_{doc \rightarrow emo}}$$

(15)

$$R_1 = \frac{\sum_{e \in E} N_{correct \rightarrow emo}}{\sum_{e \in E} N_{doc \rightarrow emo}}$$

(16)

$$Micro \cdot avgF = \frac{2 \times P_1 \times R_1}{P_1 + R_1}$$

(17)

where N_{correct \rightarrow emo} is number of documents correctly assigned to emotion class e, N_{doc \rightarrow emo} is the number of documents assigned to emotion class e, N_{doc \rightarrow emo} is the number of documents in class e. P_1 refers to micro-averaged precision, R_1 refers to micro-averaged precision recall, and E is the given set of emotions.

Macro-averaged F-score:

$$P_e = \frac{\sum_{e \in E} N_{correct \rightarrow emo}}{\sum_{e \in E} N_{doc \rightarrow emo}}$$

(18)

$$R_e = \frac{\sum_{e \in E} N_{correct \rightarrow emo}}{\sum_{e \in E} N_{doc \rightarrow emo}}$$

(19)

$$F_e = \frac{2 \times P_e \times R_e}{P_e + R_e}$$

(20)

$$Macro \cdot avgF = \frac{1}{|E|} \sum_{e \in E} F_e$$

(21)

where P_e and R_e refer to precision of each emotion and precision of each emotion in documents respectively, and E is the given set of emotions.

C. BASELINES
In order to evaluate the performance of our model, the following algorithms and models are implemented for comparison:
FastText [17]: This algorithm is a simple model proposed by Facebook AI Research, which extracts context-based semantic extraction features based on n-grams and uses context to predict the classification of text. Experiments show that the FastText algorithm can achieve the same accuracy as the deep learning model, and the computation time is smaller than the deep learning model.

CNNs [7]: The convolutional operation has the function of local feature extraction. Therefore, CNNs can be used to extract key information similar to n-gram in sentences. It can share convolution kernels and has no pressure on high-dimensional data processing. Here we choose CNN-static and CNN-non-static as baselines.

RNNs [8]: The RNNs is a neural network dedicated to processing sequence data. It can read the emotional elements contained in the text according to the word order. The key point is that the hidden state retains the previous input information and the context information.

Recurrent Convolutional Neural Networks (RCNNs) [34]: The model combines RNNs and CNNs to implement text classification task, which is a forward-backward function combine with RNNs and maximum pooling layer.

Hierarchical Neural Autoencoder (HNA) [28]: The model considers the hierarchical structure of the document: words form sentences and sentences constitute documents, so the network is divided into the word attention part and the sentence attention part when modeling, and two layers are added in the hierarchical construction.

Self-Attentive Sentence Embedding (SASE) [29]: The model is based on RNNs and combines the self-attention mechanism on the long-term memory unit. It has a significant improvement on the semantic representation of the sentence itself, which can help the model to more effectively capture words that have an important role in sentence semantics.

D. HYPER-PARAMETER DETAILS

In the training process, Sogou News word vectors [35] and Glove word vectors [36] are used as Chinese word vectors and English word vectors, respectively, and their dimensions are all 300 dimensions. In addition, we set the learning rate as 0.0003, the batch size as 128, and the number of training iterations as 50. According to our statistics on the sequence length of the data sets, the text length of tweets is padded to 50, and the text length of weibos is padded to 120, which can contain most words and avoid input matrix being too sparse. In our model and CNNs, there is only one convolutional layer, and the convolution size is set to 4, 7, and 9 to capture different levels of semantic information, and set the number of each convolutional filter to 128. In the self-attentive layer and SASE, we set the \( d_a \) to be 350 dimensions, the \( r \) to be 30 dimensions, and a ReLU output Multi-Layer Perceptron (MLP) with 2000 hidden units.

E. MODEL VARIATIONS

We experiment with two variants of the model.

- SACNN-static: Our baseline model initialize words with pre-trained word embedding and randomly initialize the unknown ones. All words keep static and only the other parameters of the model are learned.
**F. EXPERIMENTAL RESULTS**

1) EMOTION CLASSIFICATION

The results of multi-label emotion classification on test dataset of NLPCC 2014, SemEval 2018 Task 1 are listed in TABLE 2, TABLE 3. It’s obvious that our model performs best in average precision and F-score above all baselines in the multi-label emotion classification task. In addition, the weibos and tweets that do not contain emotions are used as noise variables in our comparative experiment, and add an emotion label named *none* as a flag to indicate if our data contains emotions. The results show that our model is robust, and the model performance is excellent regardless of whether noise variables are added or not. In addition, comparing the results of the model after adding noise in each experiment, we found that the proper amount of noise in the dataset may improve the classification effect of the model.

It can be observed from TABLE 2, TABLE 3, the other models perform better in Chinese corpus, and our model performs stably in both Chinese and English corpus, which proves the scalability of our proposed architecture for different corpora. HNA is not as effective as CNNs, which may be due to the fact that the hierarchical attention mechanism does not work well when the UGC is short and weibos or tweets consists of only a few short sentences. The performance of RCNNs in the two corpora is not satisfactory, which shows that the combination of the forward and backward RNNs structure and the maximum pool is not good enough to capture key emotional elements. In the absence of noise, SASE performs better than RNNs, but the SASE effect is inferior to RNNs after adding noise, which indicates that the pure RNNs structure is more robust to noise, and the self-attention mechanism is more sensitive to noise.

From the data in the TABLE 2, TABLE 3, it can be seen that the self-attention mechanism plays a better role in the noiseless situation, while the convolutional filters are less sensitive to the noise. So the combination of the self-attention mechanism and convolutional filters enable the model to achieve optimal results in UGC which contain lots of noise and multiple emotions. In addition, TABLE 2, TABLE 3 also show that our model is more sensitive to noise in Chinese corpus, and relatively insensitive to English corpus, which may be due to the ambiguity of emotion expression in Chinese language. As the results of the variations of CNNs and SACNNs show, the fine-tuning word vectors make the model perform better than the static word vector, and each variant of SACNNs performs better than that of CNNs, which shows the benefits of the self-attention mechanism.

Compared with the existing methods and experimental results, we find that the joint architecture of CNNs and self-attention mechanism models outperforms many classical text classification models in UGC. In our model structure, convolutional layer extracts the local features of the input text, and the self-attention mechanism replaces the original pooling operation, selects key local features from the convolution features and combines them, and finally performs emotional classification. Experiments confirm the validity of our ideas.

In addition, we record the data in the training process of SACNN-static and the CNN-static model, which outperform all the comparison experiment, and plot the $A_p$, Micro-F and Macro-F as a function of the number of training iterations. As can be seen from the FIGURE 5, our model reaches a higher stable value in the earlier training steps, which confirms the efficiency of our model and that it is more suitable for real-time update changes in UGC data.

2) VISUALIZATION

Our model is also able to accurately capture semantic and structural representations. The size of the convolutional filter can be adjusted to flexibly extract textual semantic information. In the following part, we give a visualization of the model to show the learned feature representation.

The self-attention mechanism is able to show the extent to which the convolution local feature representation contributes to the final classification result. A heat map is drawn to show the structure based on the attention weight and its...
corresponding filter window size. First, we extract an example from datasets of NLPCC 2014, SemEval 2018, and simulate the process of extracting features from the convolutional filter window in FIGURE 6, 7.

In FIGURE 6, FIGURE 7, we use 4, 7, and 9 as the convolutional filter sizes to extract the semantic information of different levels in the text, and associate the depth of the corresponding color with the importance of the convolution feature. Finally the key three feature windows of each convolution size are shown.

In addition, in FIGURE 8, we take a convolutional filter window of size 4 as an example, and draw a heat map based on the attention weight of the convolutional filter window, which visualizes the feature screening process and validates

FIGURE 7. Top-3 key convolutional filter windows on tweets with emotion tag: anger, sadness.

FIGURE 8. Attention weight on convolutional filter windows of size 4.

FIGURE 9. The correlation between attention weight and structure complex.

(a) The correlation between attention weight and convolutional filter window size

(b) the correlation between attention weight and sequence length
the effectiveness of our method from subjective perspective. The depth of color in FIGURE 8 represents the self-attention weight (from 0 to 1), the darker color means the higher the value, and also means the convolution feature contributes more to the final sentence representation.

3) RATIONALITY ANALYSIS

Obviously, more complex convolutional structures require higher attention weights, and longer sequences may result in lower attention weight. Two indicators are used to represent the structural complexity: the convolutional filter window size and sentence length. The FIGURE 9 depicts the correlation between normalized attention weight value and our model’s structure on datasets of NLPCC 2014 task: Chinese Weibo Text Emotion Analysis.

As can be seen from FIGURE 9, the attention weight value is positively correlated with the convolution size and negatively correlated with the sequence length. This proves the effectiveness of the self-attention mechanism in our model.

V. CONCLUSION

In this paper, we propose a SACNNs model, which combines CNNs model with the self-attention mechanism. That is, the self-attention mechanism is used to replace the pooling layer in the traditional CNNs, and the key local semantic feature information is effectively selected and merged. Finally, our model achieves higher $A_p$, Micro-F and Macro-F in the classic text classification methods, and shows better robustness. In addition, the filter size and convolutional layer of the CNNs can be adjusted and changed as needed, which indicates its powerful generalization and extensibility. In addition, we also detail the main explanations of examples, visualization and quantitative analysis.

In the future work, we will consider the global features of the text, which obtain the long-term dependency of the text, and combine the local and global semantics to improve the performance of emotion classification models.

REFERENCES

[1] C. Peter and R. Beale, Affect and Emotion in Human-Computer Interaction: From Theory to Applications (Lecture Notes in Computer Science), vol. 600, no. 1, 2008, pp. 123–125.

[2] G. Qiu, X. He, F. Zhang, Y. Shi, J. Bu, and C. Chen, “DASA: Dissatisfaction-oriented advertising based on sentiment analysis,” Expert Syst. Appl., vol. 37, no. 9, pp. 6182–6191, 2010.

[3] X. Jin, Y. Li, T. Mah, and J. Tong, “Sensitive webpage classification for content advertising,” in Proc. 1st Int. Workshop Data Mining Audience Intell. Advertising, 2007, pp. 28–33.

[4] M. I. Jordan and T. M. Mitchell, “Machine learning: Trends, perspectives, and prospects,” Science, vol. 349, no. 6245, pp. 255–260, 2015.

[5] X. Zhang, J. Zhao, and Y. LeCun, “Character-level convolutional networks for text classification,” in Proc. Adv. Neurol. Inf. Process. Syst., 2015, pp. 649–657.

[6] N. Kallabrenner, E. Grefenstette, and P. Blunsom, “A convolutional neural network for modelling sentences,” in Proc. 52nd Annu. Meeting Assoc. Comput. Linguistics, Jun. 2014, pp. 655–665.

[7] Y. Kim, “Convolutional neural networks for sentence classification,” 2014, arXiv:1408.5882. [Online]. Available: https://arxiv.org/abs/1408.5882

[8] J. B. Pollack, “Recursive distributed representations,” Artif. Intell., vol. 46, nos. 1–2, pp. 77–105, 1990.

[9] A. Vaswani, “Attention is all you need,” in Proc. Adv. Neurol. Inf. Process. Syst. 2017, pp. 5998–6008.

[10] W. Yin, “Comparative study of CNN and RNN for natural language processing,” 2017, arXiv:1702.01923. [Online]. Available: https://arxiv.org/abs/1702.01923.

[11] Y. Young, D. Hazarika, S. Poria, and E. Cambria, “Recent trends in deep learning based natural language processing,” IEEE Comput. Intell. Mag., vol. 13, no. 3, pp. 55–75, Aug. 2018.

[12] C. C. Aggarwal and C. Zhai, “A survey of text classification algorithms,” in Mining Text Data. New York, NY, USA: Springer, 2012, pp. 163–222.

[13] “A survey of recent advances in texture representation,” Tech. Rep., 2018.

[14] Q. V. Le and T. Mikolov, “Distributed representations of sentences and documents,” in Proc. Int. Conf. Mach. Learn., 2014, pp. 1188–1196.

[15] J. Zhao, Z. Zhan, Q. Yang, Y. Zhang, C. Hu, Z. Li, L. Zhang, and Z. He, “Adaptive learning of local semantic and global structure representations for text classification,” in Proc. Int. Conf. Comput. Linguistics, 2018, pp. 2033–2043.

[16] G. Salton, A. Wong, and C. S. Yang, “A vector space model for automatic indexing,” Commun. ACM, vol. 18, no. 11, pp. 613–620, 1975.

[17] A. Joulia, “Bag of tricks for efficient text classification,” in Proc. Conf. Eur. Chapter Assoc. for Comput. Linguistics, 2017, pp. 427–431.

[18] S. Gao, A. Ramanathan, and G. Tournassi, “Hierarchical convolutional attention networks for text classification,” in Proc. 3rd Workshop Represent. Learn. NLP, 2018, pp. 11–23.

[19] Y. Zhang, J. J. Marshall, and B. C. Wallace, “Rationale-augmented convolutional neural networks for text classification,” in Proc. Conf. Empirical Methods Natural Lang. Process., Nov. 2016, pp. 795–804.

[20] S. Li, “Initializing convolutional filters with semantic features for text classification,” in Proc. Conf. Empirical Methods Natural Lang. Process., Sep. 2017, pp. 1884–1889.

[21] S. Hochreiter and J. Schmidhuber, “Long short-term memory,” Neural Comput., vol. 9, no. 8, pp. 1735–1780, 1997.

[22] K. Cho, B. van Merriënboer, D. Bahdanau, and Y. Bengio, “On the properties of neural machine translation: Encoder–decoder approaches,” in Proc. 8th Workshop Syntax, Semantics Struct. Stat. Transl., Oct. 2014, pp. 103–111.

[23] R. Zhang, H. Lee, and D. R. Radev, “Dependency sensitive convolutional neural networks for modeling sentences and documents,” in Proc. North Amer. Chapter Assoc. Comput. Linguistics, Hum. Lang. Technol., Jun. 2016, pp. 1512–1521.

[24] S. Wang, J. Zhang, and C. Zong, “Exploiting word internal structures for generic Chinese sentence representation,” in Proc. Empirical Methods Natural Lang. Process., Sep. 2017, pp. 298–303.

[25] Z. Yang, D. Yang, C. Dyer, X. He, A. Smola, and E. Hovy, “Hierarchical attention networks for document classification,” in Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics, Hum. Lang. Technol., Jun. 2017, pp. 1480–1489.

[26] D. Bahdanau, K. Cho, and Y. Bengio, “Neural machine translation by jointly learning to align and translate,” in Proc. Int. Conf. Learn. Represent., May 2015, pp. 1–15.

[27] D. Hu, “An introductory survey on attention mechanisms in NLP problems,” 2018, arXiv:1811.05544. [Online]. Available: https://arxiv.org/abs/1811.05544

[28] J. Li, T. Luong, and D. Jurafsky, “A hierarchical neural autoencoder for paragraphs and documents,” in Proc. 53rd Annu. Meeting Assoc. Comput. Linguistics, 7th Int. Joint Conf. Natural Lang. Process., Jul. 2015, pp. 1106–1115.

[29] Z. Lin, M. Feng, C. N. dos Santos, M. Yu, B. Xiang, B. Zhou, and Y. Bengio, “A structured self-attentive sentence embedding,” in Proc. Int. Conf. Learn. Represent., 2017, pp. 1–15.

[30] B. Shin, T. Lee, and J. D. Choi, “Lexicon integrated CNN models with attention for sentiment analysis,” in Proc. 8th Workshop Comput. Approaches Subjectivity, Sentiment Social Media Anal., Sep. 2017, pp. 149–158.

[31] Z. Tan, “Deep semantic role labeling with self-attention,” in Proc. 32d AAAI Conf. Artif. Intell., Apr. 2018, pp. 4929–4936.

[32] Y. Zhang and B. C. Wallace, “A sensitivity analysis of (and practitioners’ guide) to convolutional neural networks for sentence classification,” in Proc. Int. Joint Conf. Natural Lang. Process., 2017, pp. 253–263.

[33] R. E. Schapire and Y. Singer, “BoosTexter: A boosting-based system for text categorization,” Mach. Learn., vol. 39, nos. 2–3, pp. 135–168, May 2000.

[34] S. Lai, “Recurrent convolutional neural networks for text classification,” in Proc. 59th AAAI Conf. Artif. Intell., Feb. 2015, pp. 2267–2273.
[35] S. Li, Z. Zhao, R. Hu, W. Li, T. Liu, and X. Du, “Analogical reasoning on Chinese morphological and semantic relations,” in Proc. 56th Annu. Meeting Assoc. Comput. Linguistics, Jul. 2018, pp. 138–143.

[36] J. Pennington, R. Socher, and C. D. Manning, “Glove: Global vectors for word representation,” in Proc. Empirical Methods Natural Lang. Process. (EMNLP), Oct. 2014, pp. 1532–1543.

YING QIAN received the B.S. and Ph.D. degrees from Kyoto University, Japan, in 1996 and 2000, respectively. He has undertaken five research projects in Japan and Canada, and is now undertaking six national, provincial, and ministerial research projects, including the National Natural Science Foundation of China and Chongqing’s major research projects. He has published dozens of articles in international and domestic journals and conferences, applied for two Japanese patents. His current research interests include information systems, digital image processing, and biomedical imaging. He was a member of the IEEE Engineering in Medicine and Biology Society, an Overseas Member of the Japan Electronics and Information Technology Association, a Senior Member of the Chinese Computer Society, a Vice Chairman of the Chinese Computer Society YOSEF (Chongqing), and a Member of the China Biomedical Engineering Society. He was a recipient of the Japan Medical Image Engineering Society Excellence Award, in 1999, the Japan Medical Image Engineering Society Excellence Award (1st Completion), the Software Export and Service Outsourcing Talent Cultivation Contribution Award (1st) of China, in 2009, and the Chongqing University of Posts and Telecommunications, Wenfeng Provincial Department Reserve Talent.

WEIWEI LIU is currently pursuing the M.S. degree with the School of Software Engineering, Chongqing University of Posts and Telecommunications, China. Her current research interests include data mining, machine learning, and natural language processing.

JIANGPING HUANG received the Ph.D. degree from Wuhan University, China, in 2017. He is currently a Lecturer with the School of Software Engineering, Chongqing University of Posts and Telecommunications, Chongqing, China. His current research interests include deep learning, text generation, paraphrase identification, emotion detection, and social computation. He has served as a PC Member for PACLIC, and a Reviewer for NLPCC, CCL, IEEE ACCESS, and ACM TKDD.