Tree-Structured Regional CNN-LSTM Model for Dimensional Sentiment Analysis

Jin Wang, Liang-Chih Yu, *Member, IEEE*, K. Robert Lai, and Xuejie Zhang

***Abstract*—Dimensional sentiment analysis aims to recognize continuous numerical values in multiple dimensions such as the valence-arousal (VA) space. Compared to the categorical approach that focuses on sentiment classification such as binary classification (i.e., positive and negative), the dimensional approach can provide a more fine-grained sentiment analysis. This article proposes a tree-structured regional CNN-LSTM model consisting of two parts: regional CNN and LSTM to predict the VA ratings of texts. Unlike a conventional CNN which considers a whole text as input, the proposed regional CNN uses a part of the text as a region, dividing an input text into several regions such that the useful affective information in each region can be extracted and weighted according to their contribution to the VA prediction. Such regional information is sequentially integrated across regions using LSTM for VA prediction. By combining the regional CNN and LSTM, both local (regional) information within sentences and long-distance dependencies across sentences can be considered in the prediction process. To further improve performance, a region division strategy is proposed to discover task-relevant phrases and clauses to incorporate structured information into VA prediction. Experimental results on different corpora show that the proposed method outperforms lexicon-, regression-, conventional NN and other structured NN methods proposed in previous studies.**

***Index Terms*—Dimensional sentiment analysis, valence-arousal prediction, regional CNN-LSTM model, structured parsing.**

I. Introduction

SENTIMENT analysis refers to the use of computational linguistics to analyze, process, induce and deduce subjective texts with affective information [1]–[5]. With the proliferation of social media content, this set of techniques raises new opportunities to study public opinion on nearly any topic. It has also been widely used in the development of online applications for customer reviews analysis [6], mental illnesses identification [7], hotspot detection and forecasting [8], question answering [9], social text analysis [10], [11] and financial market prediction [12].

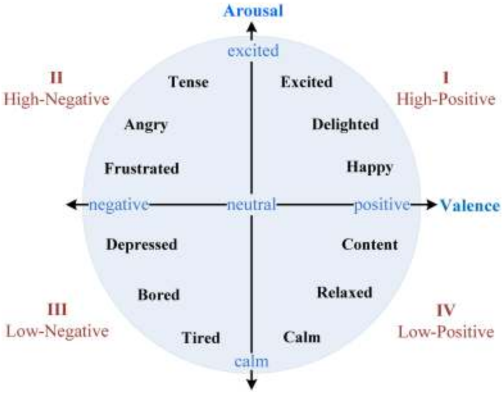


Fig. 1. Two-dimensional valence-arousal space.

Research related to affective computing theory provides two approaches to represent an emotional state: categorical and dimensional [13]. The categorical approach represents emotional states as several discrete classes such as binary (i.e., positive and negative) or as multiple categories such as Ekman's six basic emotions (anger, happiness, fear, sadness, disgust, and surprise) [14] and Plutchik's eight emotions (Ekman's six plus trust and anticipation) [15]. Classification algorithms can then be used to identify sentiment categories from texts. However, beyond the pre-defined emotion categories, this approach cannot describe more fine-grained differences between emotions.

The dimensional approach represents emotional states as continuous numerical values in multiple dimensions to reflect differences in sentiment strength or intensity. As shown in Fig. 1, the most commonly used approach is based on the valence-arousal space proposed by Russell *et al.* [16], which can accurately represent the affective state in a 2-dimentional continuous space. In this space, the dimension of valence refers to the degree of positive and negative sentiment, whereas the dimension of arousal refers to the degree of calm and excitement. Both dimensions range from 1 (highly negative or calm) to 9 (highly positive or excited) based on the self-assessment manikin (SAM) annotation scheme [17]. By using such a representation, any emotional state in a subjective text can be marked as a point in the valence-arousal coordinate plane. For example, the following three-sentence passage is associated with a valence-arousal rating of (2.5, 7.8), indicating a high degree of negativity and arousal.

Manuscript received April 19, 2019; revised September 26, 2019; accepted November 23, 2019. Date of publication December 11, 2019; date of current version January 21, 2020. This work was supported in part by the National Natural Science Foundation of China under Grants 61702443, 61966038, and 61762091, and in part by the Ministry of Science and Technology, Taiwan, ROC, under Grants MOST 107-2628-E-155-002-MY3, and MOST 107-2218-E-006-008. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Eric Fosler-Lussier. This paper was presented in part at the Fifty fourth Annual Meeting on Association for Computational Linguistics, Berlin, Germany, 2016, pp. 225–230. *(Corresponding author: Liang-Chih Yu.)*

J. Wang and X. Zhang are with the School of Information Science and Engineering, Yunnan University, Kunming 650000, China (e-mail: wangjin@ynu.edu.cn; xjzhang@ynu.edu.cn).

L.-C. Yu is with the Department of Information Management, Yuan Ze University, Taoyuan 32003, Taiwan (e-mail: lcyu@saturn.yzu.edu.tw).

K. R. Lai is with the Department of Computer Science and Engineering, Yuan Ze University, Taoyuan 32003, Taiwan (e-mail: krlai@saturn.yzu.edu.tw).

Digital Object Identifier 10.1109/TASLP.2019.2959251

(r1) A few days ago I checked into a franchise hotel.

(r2) The front desk service was ***terrible****, and they* ***didn't know*** *much about local attractions.*

(r3) I would ***not recommend*** *this hotel to a friend.*

Such high-arousal negative (or high-arousal positive) texts are usually of interest and could be prioritized in product review systems. Dimensional sentiment analysis can accomplish this by recognizing the VA ratings of texts and rank them accordingly, thus providing more intelligent and fine-grained sentiment applications.

Recently, word embedding [18]–[21] and deep neural networks (NN) such as convolutional neural networks (CNN) [22], [23], recurrent neural networks (RNN) [24], [25], gated recurrent unit (GRU) [26], and long short-term memory (LSTM) [27], [28] have been successfully employed for categorical sentiment analysis. Given a variable-length text, one challenge of using these neural networks is to compose individual word vectors into sentence vectors of the same length for polarity classification. One challenge of using these methods for VA prediction lies in how to distill sentiment information from the salient words by assigning more weight which contribute to the final prediction. For example, the overall affective states in the aforementioned passage are mainly determined by the bolded words, such as *terrible*, *didn't know* and *not recommend*.

To use information from words, one intuitive method is to directly average all word vectors in each dimension [29]–[31]. Unfortunately, every word vector in these methods shared an equal weight, making it very similar to the concept of Bag-of-Words (BoW). In contrast, CNN uses a convolutional kernel to extract local *n*-gram features and then max-pooling to select the most salient features for prediction. However, this often misses valuable information present in multiple facts within a very long sentence, and may fail to capture long-distance dependency. RNN, GRU and LSTM can address this limitation by sequentially modeling texts across sentences. However, RNN, GRU and LSTM are bias models, where the key components in the tail of a text dominate the key components in the header, resulting in the model always making decisions based on tail information. To capture both local and long-distance information, an LSTM layer can be combined with a CNN layer to form a CNN-LSTM model. Such NN-based and word embedding methods have not been well explored for dimensional sentiment analysis.

This study proposes a regional CNN-LSTM model consisting of two parts, regional CNN and LSTM, to predict the VA ratings of texts. We first construct word vectors for vocabulary words using word embedding. The regional CNN is then used to build text vectors for the given texts being predicted based on the word vectors. Unlike a conventional CNN which considers a whole text as input, the proposed regional CNN uses a part of the text as regions, dividing an input text into several regions such that the key components and useful affective information in different regions can be extracted and weighted according to their contribution to the VA prediction. For example, in the aforementioned example text, it would be useful for the system to emphasize the two sentences/regions (*r*2) and (*r*3) containing negative affective information. Finally, such regional information is sequentially integrated across regions using LSTM for VA prediction. 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.

To better discover the implicit structure and extracting local salient features, we also propose a tree-structured region division strategy. By using a constituency-based tree parser, a given text can be divided into regions according to different tree depths. These regions are linguistic function blocks, which could be words, phrases, clauses, sentences, or even paragraphs. In each region, the proposed regional CNN-LSTM will extract the appropriate key components and learn the linguistic relations between them to contribute to VA prediction. By using the parser to identify regions, the structural information can be incorporated to improve prediction performance.

Comparative experiments were conducted on four English and Chinese corpora with annotated valence-arousal values. We first investigate the effect of the regional division on different tree depths of the tree-structured regional CNN-LSTM model. The experimental results show that the proposed tree-structured regional CNN-LSTM model outperformed several existing methods, such as lexicon-, regression-, and conventional NN-based methods.

The rest of this paper is organized as follows. Section II introduces the existing methods for predicting VA ratings of affective texts. Section III describes the proposed tree-structured regional CNN-LSTM model and also the strategy of regional division on a parsed tree representation. Section IV reports the evaluation results of the proposed method against lexicon-based, regression-based, conventional NN and structured NN methods. Conclusions are finally drawn in Section V.

II. Related Works

Dimensional sentiment analysis in VA space can provide more fine-grained affective information for other sentiment applications than the traditional categorical approach. In this section, we present a brief review of existing text VA prediction methods, including lexicon-, regression-, and conventional neural network-based methods.

A. Lexicon-Based Methods

Lexicon-based methods assume that a text's affective ratings can be estimated via the composition of the affective scores of its component words. To predict the affective rating of each text, these methods use an affective lexicon in which affective words are tagged with valence and arousal ratings, such as Affective Norms of English Words (ANEW) [32].

Given the affective scores of words, one may calculate the affective scores of a text through different composition methods. A feasible method for composition is arithmetic mean. That is, VA values of a text *t* can be predicted by the average VA values of each word *w* in this sentence, defined as,

$$val_t = \frac{1}{n}\sum_{w\in t} val_w \tag{1}$$

where val_t and val_w respectively denote the valence values of sentence *t* and word *w*.

Instead of simply using the arithmetic mean affective values of words, Paltoglou *et al.* [10] used three different methods to estimate a text's overall sentiment score, including weighted arithmetic mean, weighted geometric mean, and Gaussian mixture model.

- Weighted arithmetic mean (*w*AM)

$$val_t = \sum_{w\in t} tf_w \times val_w \Big/ \sum_{w\in t} tf_w \tag{2}$$

  where tf_w is the term frequency of word *w* in text *t*.
- Weighted geometric mean (*w*GM)

$$val_t = \sqrt[\sum_{w\in t} tf_w]{\prod\nolimits_{w\in t} (val_w)^{tf_w}} \tag{3}$$

- Gaussian mixture model (GMM)

$$val_t = \arg\max_x \left\{ \sum_{w\in t} N(x|\mu_w, \sigma_w^2) \times tf_w \Big/ \sum_{w\in t} tf_w \right\} \tag{4}$$

  where $N(x \mid \mu_w, \sigma^2 w)$ is the probability density function of word *w*, following the Gaussian distribution with mean μ_w and variance $\sigma^2 w)$, which can be derived from the ANEW lexicon. Experimental results show that the weighted geometric mean method outperforms other two methods.

Although such methods can be easily implemented, they cannot model a text or document with complex linguistic expressions. For example, if a negative review contains more positive words than negative words, it will be incorrectly classified as positive, thus the emotional import of a text or document is not simply the sum of emotional associations of its constituent words.

B. Regression-Based Methods

Regression-based methods have been intensively studied for VA prediction at both the word level [33]–[35] and the sentence level [10], [11], [36]. At the word level, Wei *et al.* [33] used linear regression to transfer VA ratings from English affective words to Chinese words. Malandrakis *et al.* [34] used a kernel function to combine the similarity between words for VA prediction. Yu *et al*. [35] and Wang *et al*. [37] and used a weighted graph model to iteratively determine the VA ratings of affective words.

At the sentence level, Gokcay *et al.*, [38] proposed applying a linear regression model on an affective lexicon to predict the overall sentiment score of texts. The candidate texts are decomposed into their words to lookup sentiment scores from an affective lexicon. A list of stop-words is used to remove words that are not found in the lexicon. The sentiment scores of the text and the average scores of words in the text are then taken as input to train a regression model.

Instead of simply using the mean affective values of words, Malandrakis *et al.* [36] extracted *n*-grams, the weighted mean and maximum affective values of constituent words as features to train regression models. The authors also proposed a method that extracts *n*-grams with affective ratings as features to predict sentiment VA values for sentences and complete texts.

Paltoglou and Thelwall [11] predicted valence and arousal levels of a sentence or document on an ordinal five-level scale, from very negative/low to very positive/high. The authors considered the sentiment prediction problem as both a classification and regression. Both methods are based on BoW features. Support vector machine (SVM) and ε-support vector regression (ε-SVR) are then respectively used for classification and regression. Their experimental results also show that regression techniques tend to produce smaller scale errors.

C. Conventional Neural Network Methods

When used as the input representation of a learning system, word and phrase embeddings have been shown to boost performance in several natural language processing (NLP) tasks [18], [19]. In composing individual word vectors into sentence vectors, deep average networks (DAN) are commonly used to average the word vectors in a given text [29]–[31]. However, this approach loses word order information, making it practically equivalent to the concept of BoW.

Other NN models for polarity classification include CNN [22], [23], RNN [24], [25], GRU [26] and LSTM [27], [28]. In a basic CNN, a local *n*-gram tensor is convolved with a set of kernels. To minimize computational complexity, CNN uses max pooling which reduces the size of the output from the previous convolutional layer. RNN is a powerful method for processing text, string, and sequential data. The RNN architecture considers the information of previous words in a very sophisticated method which allows for better representation of texts with sequence order information. However, simple RNNs pose training challenges due to vanishing and exploding gradient problems. To address this issue, LSTM and GRU use multiple gates to carefully regulate the amount of information allowed into each hidden state. They preserve long term dependency more effectively than basic RNNs.

These deep NN models can be further extended by stacking multiple layers. One intuitive way is to use a hierarchical CNN-LSTM model [39]. This model successively stacks a CNN layer, a max-pooling layer and an LSTM layer, so that the model can consider both local *n*-gram features and long dependencies. It does not apply region divisions, where LSTM is still performed on the sequence of the feature maps output by the max-pooling layer.

D. Structured Representation Models

Conventional NN models only consider word order or local dependency, rather than any structure information of text. To learn structured representation, the tree-structured LSTM model [40], [41] and recursive auto-encoder [42], [43] are proposed to apply pre-specified parsing trees to build structured representations via a recursive RNN or LSTM units. However, such models

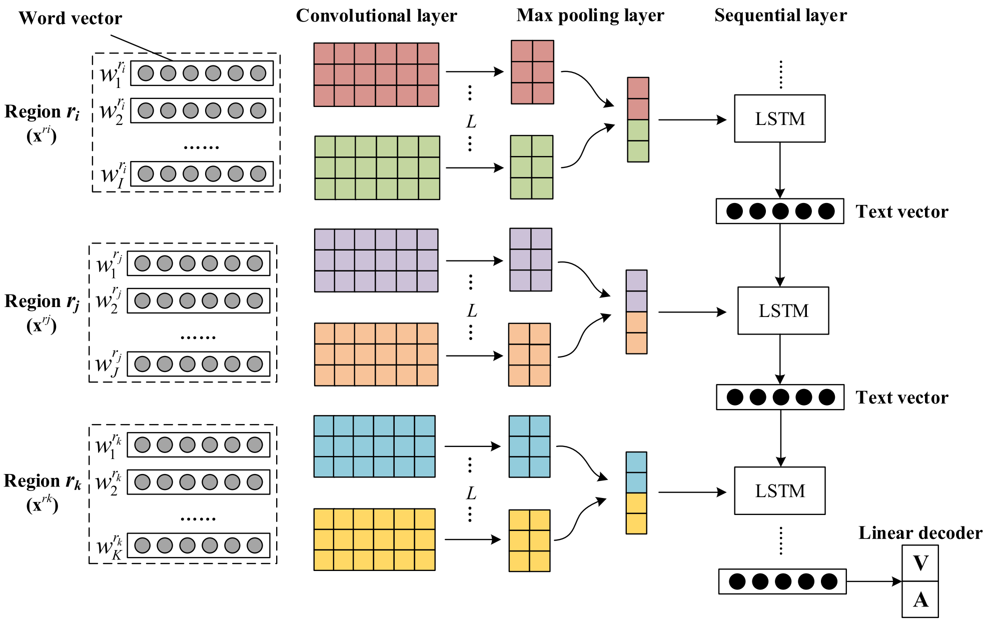


Fig. 2. System architecture of the proposed regional CNN-LSTM model.

could not avoid the bias problem, since the words that contribute to the prediction but lie in the leaf node at the bottom of the parse tree will be less emphasized.

To highlight the keywords in prediction, a self-attention mechanism [44], [45] was used to extract domain-related information. It can automatically learn the contribution of each hidden state, and assign more weight to high contributors.

III. Tree-Structured Regional CNN-LSTM Model

The procedure for using a tree-structured regional CNN-LSTM model for VA prediction consists of two parts: the regional CNN-LSTM model and a regional division strategy. Instead of using the whole text as input, the regional CNN-LSTM model divides each input text into several regions to extract both local n-gram features within regions and long-distance dependencies between regions. To better divide regions, a tree-structured region division strategy is introduced such that the linguistic structures at different tree depths (regions) can be incorporated into the prediction process. The following subsections provide a detailed explanation of the architecture of the regional CNN-LSTM model and regional division strategy.

A. *Regional CNN-LSTM Model*

Fig. 2 shows the overall framework of the proposed regional CNN-LSTM model. First, the word vectors of the vocabulary words are trained from a large corpus using the word vector learning toolkit. For each given text, the regional CNN model uses a part of the given text as a region to divide the text into R regions, i.e., $r_1, \ldots, r_i, r_j, r_k, \ldots, r_R$. In each region, useful affective features can be extracted once the word vectors sequentially pass through a convolutional layer and max pooling layer. Such local (regional) features are then sequentially integrated across regions using LSTM to build a text vector for VA prediction.

1) Convolutional Layer: In each region, a convolutional layer is first used to extract local n-gram features. All word embeddings are stacked in a region matrix $M \in \mathbb{R}^{d\times|V|}$, where $|V|$ is the vocabulary size of a region, and d is the dimensionality of the word vectors. For example, in Fig. 1, the word vectors in the regions $r_i = \{w_1^{ri}, w_2^{ri}, \ldots, w_I^{ri}\}$, $r_j = \{w_1^{rj}, w_2^{rj}, \ldots, w_J^{rj}\}$ and $r_k = \{w_1^{rk}, w_2^{rk}, \ldots, w_K^{rk}\}$ are combined to form the region matrices $\mathbf{x}^{ri}$, $\mathbf{x}^{rj}$, and $\mathbf{x}^{rk}$. In each region, we use L convolutional kernels to learn local n-gram features. In a window of ω words $\mathbf{x}_{n:n+\omega-1}$, a kernel F_l $(1 \leq l \leq L)$ generates the feature map y_n^l as follows,

$$y_n^l = f(W^l \circ \mathbf{x}_{n:n+\omega-1} + b^l) \qquad (5)$$

where $\circ$ is a convolutional operator, b^l denotes the weight matrix and bias associated with the kernel F_l, ω is the length of the kernel, d is the dimension of the word vector, and f is the ReLU function. When a kernel gradually traverses from $\mathbf{x}_{1:\omega-1}$ to $\mathbf{x}_{N+\omega-1:N}$, we get the output feature maps $\mathbf{y}^l = y_1^l, y_2^l, \ldots, y_{N-\omega+1}^l$ of kernel F_l. Given varying text lengths in the regions, $\mathbf{y}^l$ may have different dimensions for different texts. Therefore, we define the maximum length of the CNN input in regions as the dimension N. If the input length is shorter than N, then zero vectors will be appended. As shown in Fig. 2, each convolutional layer takes as its input a region vector to L different kernels with different colors, and produces feature maps $\mathbf{Y} = \{\mathbf{y}^1, \mathbf{y}^2 \ldots, \mathbf{y}^L\} \in \mathbb{R}^{(N-\omega+1)\times L}$.

2) Max-Pooling Layer: Max-pooling subsamples the output of the convolutional layer. The most common way to perform pooling is to apply a max operation with a pooling size s to the result of each kernel. The max-pooling layer can extract the local dependency within different regions to keep the most salient information. The obtained region matrix is flattened to a vector and then fed to the sequential layer.

3) Sequential Layer: To capture long-distance dependencies across regions, the sequential layer sequentially integrates each region vector into a text vector. Here, LSTM is introduced in the sequential layer for vector composition. After the LSTM cell successively traverses through all regions, the last hidden state of the sequential layer is regarded as the text representation **t** for VA prediction.

4) Linear Decoder: Since the values in both the valence and arousal dimensions are continuous, the VA prediction task requires a regression. Instead of using a softmax classifier, a linear activation function (also known as a linear decoder) is used in the output layer, defined as,

$$\begin{aligned} val_t &= W_{val}\mathbf{t} + b_{val} \\ aro_t &= W_{aro}\mathbf{t} + b_{aro} \end{aligned} \tag{6}$$

where **t** is the text vector learned from the sequential layer, val_t and aro_t is the degree of valence and arousal of the target text. *W* and *b* respectively denote the weight and bias associated with the linear decoder.

The regional CNN-LSTM model is trained by minimizing the mean squared error between the predicted *y* and actual *y*. Given a training set of text matrix $\mathbf{X} = \{\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \ldots, \mathbf{x}^{(m)}\}$, and their VA ratings set $\mathbf{y} = \{y^{(1)}, y^{(2)}, \ldots, y^{(m)}\}$, the loss function is defined as

$$L(\mathbf{X}, \mathbf{y}) = \frac{1}{2m} \sum_{i=1}^{m} \left\| h(\mathbf{x}^{(i)}) - y^{(i)} \right\|^2 \tag{7}$$

In the training phase, a back propagation (BP) algorithm with Adam optimizer is used to learn the model parameters. Details of the BP algorithm can be found in [46].

B. Region Division Strategy

In the regional CNN-LSTM model, region size will determine the range of the convolutional layer to extract key component features. Therefore, a reasonable regional division strategy will ultimately affect prediction performance. This study proposes two regional division strategy: *sequential* and *tree-structured* approaches.

1) Sequential Division Strategy: One simple way is to take a sequential approach, which considers each individual sentence in the text as a region. For instance, if a given text contains three sentences, they will be assigned to three regions. In each region, an individual convolutional and max-pooling layer is applied to extract the most important information, which is then input into a global sequential layer containing three LSTM recurrent units. Although this strategy is easy to implement, it will be very imbalanced given a large sentence length margin. Since both a very long and very short sentence will be assigned to a single region, the key component features in the long sentence will be more difficult to extract.

2) Tree-Structured Division Strategy: An alternative way is to parse the given text as a tree-structured topology, which better represents text meaning than the sequential approach. Based on the parse tree, a given text can be divided into regions according to different tree depths. Such regions could represent linguistic expression function blocks, such as words, phrases, clauses, sentences, or even an entire paragraph.

Fig. 3 uses a set of diagrams to explain the idea of dividing regions based on a parser tree. Taking nodes at different depths of the parser tree as the regional roots, the associated child nodes are grouped into one region. The proposed method can benefit from this division strategy to discover different linguistic structures at different levels of granularity. For instance, in Fig. 3(a), the rectangle marked with the dotted lines represents the range of a region. By taking the node (depth = 1) as the root, the regional CNN-LSTM model groups the whole text as a single region, and is performed similar to a single CNN model with an LSTM unit. As shown in Figs. 3(b) and Fig. 3(c), the nodes (respectively depth = 2 and depth = 3) are selected as the regional roots. Their child nodes are grouped to form regions which will be then input into the convolutional layer. In addition, as the depth increases, the number of tokens decreases in each region. Each region can thus capture finer linguistic structures (e.g., clauses and phrases). As shown in Fig. 3(d), when the maximum depth of the tree was selected as the division criterion, every node will be considered as the regional root. As a result, each region will contain only a single word. The convolution that operates on one word will become a non-linear transform for the word vector, which is then input into the LSTM layer. In this circumstance, the regional CNN-LSTM model performs similarly to an LSTM model.

Compared to the sequential approach that takes individual sentences as regions, the tree-structured approach can dynamically change the model structure to extract key components at different depths according to the tree-structured topology.

IV. Experiments

This section first investigates the effect of the regional division on different tree depths of the tree-structured regional CNN-LSTM model on prediction performance. We then evaluate the performance of the proposed model against lexicon-, regression- , and conventional NN-based methods.

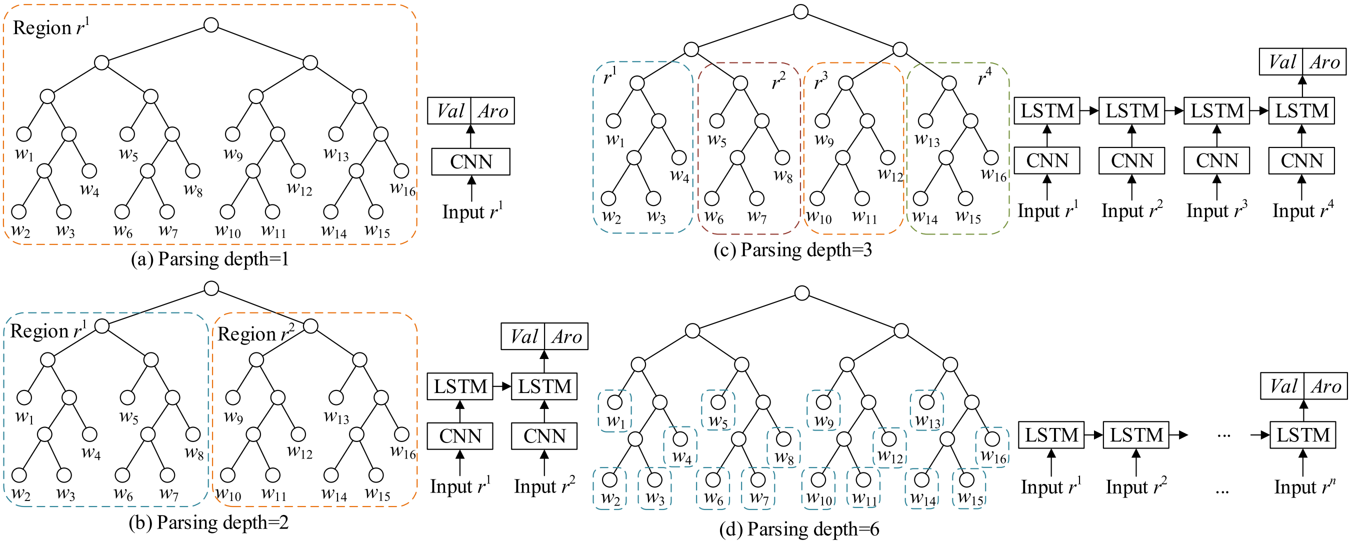
A. Dataset

This experiment used four affective corpora:

- **Stanford Sentiment Treebank**[1] **(SST)** [42] contains 8,544 training texts, 2,210 test texts, and 1,101 validation texts. Each text is rated with a single dimension (valence) in the range of (0, 1). Although in most cases the SST is used for classification tasks, it can still be used in regression tasks if the user-annotated sentiment intensity is used.
- **EmoBank**[2] [47], [48] corpus comprises 10,240 sentences with VA annotation also using the SAM annotation scheme [17]. Each sentence in EmoBank was annotated according to both the emotion expressed by the writer, and the emotion perceived by the reader.

[1] [Online]. Available: https://nlp.stanford.edu/sentiment/treebank.html

[2] [Online]. Available: https://github.com/JULIELab/EmoBank



(a) Parsing depth=1

(b) Parsing depth=2

(c) Parsing depth=3

(d) Parsing depth=6

Fig. 3. Example of region division according to the different depths of a tree-structured representation.

- **The Valence and Arousal Facebook Posts (FB)** [49] contains 2895 social media posts rated with valence and arousal values by two psychologically-trained annotators
- **Chinese Valence-Arousal Texts**[3] **(CVAT)** [50] consists of 2,009 texts collected from social forums, manually rated with both valence and arousal dimensions in the range of (1, 9) also using SAM.

EmoBank (both reader and writer), FB and CVAT were randomly split into training, development and test sets using a 7:2:1 ratio for 5-fold cross-validation. The word vectors for English and Chinese were respectively trained using the 840B Common Crawl and Chinese wiki dumps[4] (zhwiki) datasets trained by GloVe toolkit.[5] The dimensionality for both word vectors is 300.

B. Experimental Settings

The tree-structured regional CNN-LSTM model was implemented for comparison with several existing methods, such as lexicon-, regression- and conventional NN-based methods. The implementation details for each method are described as follows.

- ***w*AM and *w*GM**: Weighted arithmetic mean (*w*AM) and weighted geometric mean (*w*GM) are both lexicon-based methods [10]. In these methods, the valence/arousal values of a given text are estimated via the weighted mean of the affective values of tokens in the text.
- **AVR and MVR**: Average values regression (AVR) and maximum values regression (MVR) extract the weighted and maximum valence/arousal value of constituent words as features to train regression models [36].
- **CNN, RNN, and LSTM**: Three conventional NN methods are introduced for comparison: CNN [22], [23], RNN [24], [25] and LSTM [27], [40]. To enhance the performance of the GRU and LSTM layers, we introduce a bi-directional strategy [51]. At each time step, the hidden state of the bidirectional LSTM is the concatenation of the forward and backward hidden states to capture both past and future information.
- **Attention LSTM**: The self-attention mechanism is usually used to improve the performance of LSTM or GRU by automatically learning the contribution of each hidden state, and assigning more weight to high contributors [44], [45].
- **Tree-Structured LSTM**: Tree-LSTM [40], [41] uses prespecified parsing trees to build structured representations via a recursive LSTM unit. This model is expected to capture the structured information which affects the subsequent prediction.
- **CNN-LSTM**: This method successively stacks a CNN layer, a max-pooling layer and an LSTM layer to form a CNN-LSTM model [39]. This model does not divide regions, so that LSTM is performed on the sequence of the feature maps output by the max-pooling layer.
- **Two-Layer Attention LSTM (HAN):** Hierarchical Attention Networks (HAN) implement two LSTM layers with attention mechanisms applied at the word- and sentence-level, to pay more or less attention to individual words and sentences when composing final text representation [45]. The model takes each sentence as a region.
- **Regional CNN-LSTM**: The proposed method is implemented using the sequential and tree-structured division strategies presented in Section III.B. The sequential approach, denoted as Regional CNN-LSTM (Sequential), considers each individual sentence in the text as a region, similar to Two-layer Attention LSTM (HAN). The tree-structured approach, denoted as Regional CNN-LSTM (Tree), takes nodes at different depths of the parse tree as the division criterion.

In the above methods, the valence-arousal ratings of English and Chinese words were respectively taken from the Extended ANEW [52] and Chinese Valence-Arousal Words (CVAW) lexicons [50]. For the NN models, we introduce spatial dropout

[3] [Online]. Available: http://nlp.innobic.yzu.edu.tw/resources/cvat.html
[4] [Online]. Available: https://dumps.wikimedia.org/
[5] [Online]. Available: https://nlp.stanford.edu/projects/glove/

TABLE I
HYPER-PARAMETERS USED IN EACH CLASSIFIERS

Methods	CNN	RNN, LSTM	CNN-LSTM
Filter Number	60	-	60
Filters Length	3	-	3
Pool Length	2	-	2
Hidden State Dim.	–	120	120
Optimizer	Adam		
Batch Size	32		
(Recurrent) Dropout	0.25		
Epoch	20		

TABLE II
STATISTICAL RESULTS OF PARSER TREE AND OPTIMAL DEPTHS ON DIFFERENT DATASETS

Corpus	Tokens			Parse Tree (depth)			
	Max.	Mean	Std.	Max.	Mean	Opti. (Val)	Opti. (Aro)
SST	55	19.4	9.2	44	14.2	5	-
EB (reader)	134	17.6	12.7	44	8.3	4	4
EB (writer)	134	17.6	12.7	44	8.3	4	5
FB	475	19.7	19.9	61	9.4	5	5
CVAT	140	33.1	22.9	39	12.8	8	8

TABLE III
RESULTS OF TREE-STRUCTURED REGIONAL CNN-LSTM WITH OPTIMAL PARAMETERS ON THE DEVELOPMENT SET AND TEST SET OF DIFFERENT DATASETS

Corpus		Dev. Set			Test Set		
		Num.	MAE	*r*	Num.	MAE	*r*
SST	Int	2,210	0.952	0.788	1,101	0.943	0.809
EB (reader)	Val	1,024	0.508	0.622	2,048	0.496	0.626
	Aro		0.430	0.509		0.432	0.504
EB (writer)	Val	1,024	0.412	0.564	2,048	0.408	0.566
	Aro		0.452	0.492		0.448	0.499
FB	Val	289	0.634	0.720	578	0.627	0.726
	Aro		0.635	0.899		0.623	0.905
CVAT	Val	201	0.850	0.786	402	0.823	0.793
	Aro		0.659	0.577		0.645	0.580

(in CNN) and recurrent dropout (in LSTM) as a means of regularization to prevent overfitting problems. The dropout rate was set to 0.25. All NN methods are implemented using the Gensim,[6] TensorFlow,[7] and Keras[8] toolkits, with default parameter settings, as summarized in Table I.

C. Evaluation Metrics

Performance was evaluated using the Pearson correlation coefficient (*r*) and mean absolute error (MAE), defined as follows,

- Pearson correlation coefficient (*r*)

$$r = \frac{1}{n-1}\sum_{i=1}^{n}\left(\frac{a_i-\mu_A}{\sigma_A}\right)\left(\frac{p_i-\mu_P}{\sigma_A}\right) \tag{8}$$

- Mean absolute error (MAE)

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|a_i-p_i| \tag{9}$$

where $a_i \in A$ and $p_i \in P$ respectively denote the *i*-th actual value and predicted value, *n* is the number of test samples, μ_A and σ_A represent the mean value and the standard deviation of *A*, while μ_P and σ_P represent the mean value and the standard deviation of *P*. The MAE results reflect the difference between the predicted values of sentiment intensities and the corresponding manually rated actual values in the four corpora. The Pearson correlation coefficient is a measure of the linear correlation between the actual value and the predicted value. A higher *r* and a lower MAE value indicate better prediction performance. A Whitney-Mann *u*-test was used to determine whether the performance difference was statistically significant.

D. Regional Division Selection

As previously described, nodes at different depths of the parse tree are used as the criterion for regional division. A reasonable division strategy will ultimately improve prediction performance. The optimal settings of the division depth were determined using the development set of all VA datasets. Fig. 4 and 5 respectively show the prediction performance, including MAE and *r*, against different division depth settings using the tree-structured regional CNN-LSTM model for EmoBank (both writer and reader), FB, CVAT and SST. The results presented in Fig. 4(a) show that the optimal depth of the regional division was at the fourth layer of the parse tree on EmoBank (reader) for both valence and arousal. Once the optimal value is exceeded, performance gradually decreases because the regions shrink in size to eventually contain only a single word, indicating that properly controlling the regional division contributes to the final affective rating prediction.

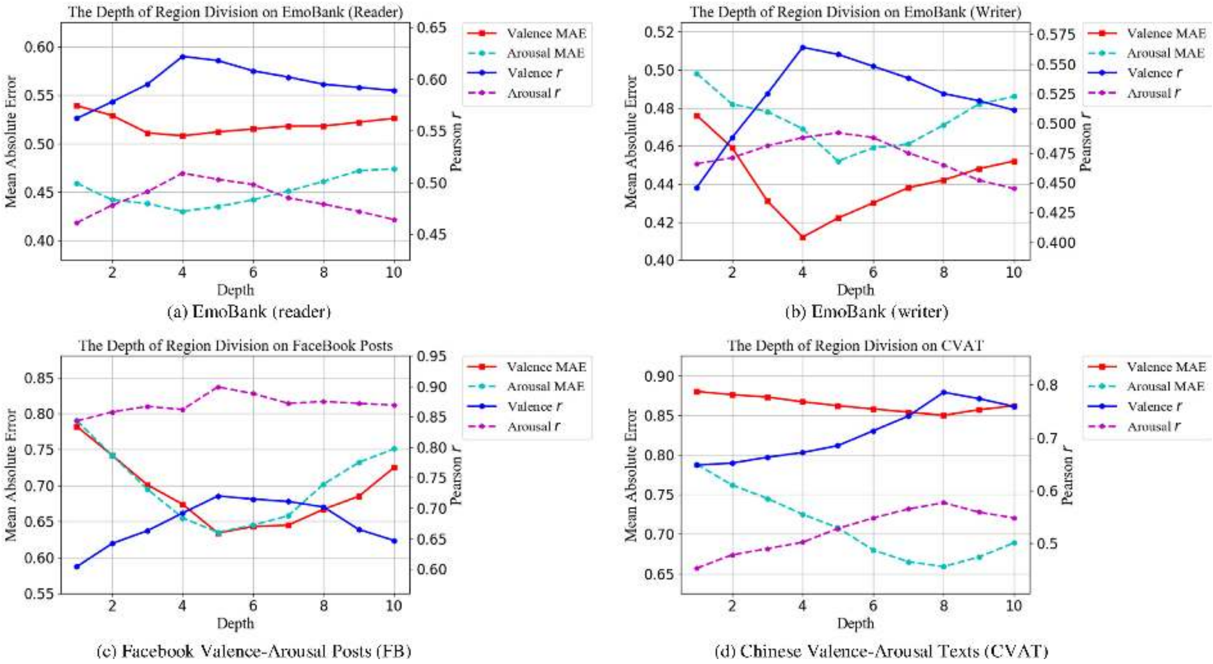
In addition, the depth of regional division in the parse tree is closely related to the length of the text. Table II summarizes the maximum and mean numbers of tokens, and also the maximum and mean depth for regional division of the parse trees. As indicated, the optimal depth of division leaves each region containing 4 or 5 words, implying phrases and clauses which would provide more affective and structural information conducive to the final prediction. The detailed analysis of the structure in region is presented in following section.

Once the optimal settings of regional division depth were obtained using the development set of SST, EmoBank (both reader and writer), FB and CVAT, they are respectively used for prediction with the associated test sets. Table III shows the performance of the proposed tree-structured regional CNN-LSTM with the optimal division depth on the development set (Figs. 4 and 5) and the test set of SST, EmoBank (both reader and writer), FB and CVAT, showing the performance on the test set was very close to that on the development set for all datasets.

[6] [Online]. Available: http://radimrehurek.com/gensim/

[7] [Online]. Available: http://www.tensorflow.org/

[8] [Online]. Available: https://keras.io/



(a) EmoBank (reader)

(b) EmoBank (writer)

(c) Facebook Valence-Arousal Posts (FB)

(d) Chinese Valence-Arousal Texts (CVAT)

Fig. 4. Region division selection for the tree-structured regional CNN-LSTM model.

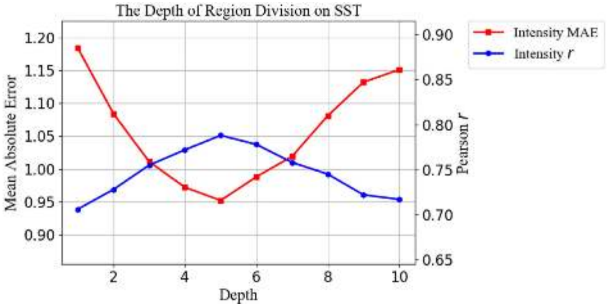


Fig. 5. Regional Division Selection for the tree-structured regional CNN-LSTM model on SST.

TABLE IV
COMPARATIVE RESULTS OF DIFFERENT METHODS IN SST

Sentiment Intensity (Valence)		SST	
		MAE	*r*
Lexicon	Lexicon-*w*AM	1.709	0.350
	Lexicon-*w*GM	1.692	0.385
Regression	Regression-AVR	1.542	0.455
	Regression-MVR	1.551	0.448
Conventional NN	CNN	1.184	0.706
	RNN	1.715	0.401
	LSTM	1.151	0.717
Structured NN	Attention LSTM	1.021	0.755
	Tree-LSTM	1.019	0.763
	CNN-LSTM	1.025	0.733
	Two-layer Attention LSTM (HAN)	0.976	0.782
	Regional CNN-LSTM (Sequential)	0.987	0.778
	Regional CNN-LSTM (Tree)	**0.943**	**0.809**

*Regional CNN-LSTM (Tree) vs. Two-layer Attention LSTM (HAN) differ significantly ($p < 0.05$).

E. Comparative Results

Tables V and VI respectively present the comparative results of the regional CNN-LSTM against several methods for VA prediction of texts in both the English and Chinese corpora. For the lexicon-based methods, *w*GM outperformed *w*AM, which is consistent with the results presented in [10]. Instead of using the VA ratings of words to directly measure those of texts, the regression-based methods learned the correlations between the VA ratings of words and texts, thus yielding better performance. Introducing the word embedding and deep learning techniques dramatically improved the performance of NN-based methods (except for RNN). In the conventional NN model, LSTM outperforms CNN and RNN due to its ability to represent a text with sequence order information and long-distance dependencies. By introducing a self-attention mechanism or tree-structure information, both Attention LSTM and Tree-LSTM outperformed the LSTM model. For the two-layer architecture, Two-layer Attention LSTM (HAN) outperformed CNN-LSTM and its results were similar to those of Regional CNN-LSTM (Sequential) because both Two-layer Attention LSTM (HAN) and Regional CNN-LSTM (Sequential) use each sentence as a region. The proposed Regional CNN-LSTM (Tree) outperformed Two-layer Attention LSTM (HAN) with a statistically significant performance difference ($p < 0.05$). This indicates that incorporating the structural information at different tree depths through region division can further improve prediction performance.

Another observation is that the Pearson correlation coefficient of prediction in arousal is lower than that for the valence prediction, indicating that arousal is more difficult to predict.

TABLE V
COMPARATIVE RESULTS OF DIFFERENT METHODS IN DIFFERENT CORPORA WITH VALENCE RATINGS

Valence		EmoBank (reader)		EmoBank (writer)		FB		CVAT	
		MAE	*r*	MAE	*r*	MAE	*r*	MAE	*r*
Lexicon	Lexicon-*w*AM	0.782	0.443	0.695	0.342	0.892	0.477	1.632	0.406
	Lexicon-*w*GM	0.779	0.445	0.692	0.347	0.895	0.472	1.597	0.418
Regression	Regression-AVR	0.759	0.462	0.684	0.354	0.854	0.486	1.374	0.476
	Regression-MVR	0.765	0.458	0.675	0.358	0.852	0.489	1.392	0.468
Conventional NN	CNN	0.587	0.561	0.476	0.446	0.782	0.604	0.880	0.645
	RNN	0.639	0.523	0.530	0.437	0.788	0.542	1.262	0.493
	LSTM	0.523	0.589	0.452	0.511	0.725	0.631	0.869	0.647
Structured NN	Attention LSTM	0.502	0.609	0.428	0.542	0.718	0.640	0.845	0.777
	Tree-LSTM	0.517	0.597	0.441	0.524	0.721	0.638	0.843	0.780
	CNN-LSTM	0.518	0.592	0.449	0.520	0.716	0.643	0.852	0.685
	Two-layer Attention LSTM (HAN)	0.507	0.604	0.424	0.557	0.667	0.691	0.839	0.778
	Regional CNN-LSTM (Sequential)	0.507	0.602	0.434	0.532	0.690	0.666	0.842	0.781
	Regional CNN-LSTM (Tree)	**0.496**	**0.626**	**0.408**	**0.566**	**0.627**	**0.726**	**0.823**	**0.793**

*Regional CNN-LSTM (Tree) vs. Two-layer Attention LSTM (HAN) differ significantly ($p < 0.05$).

TABLE VI
COMPARATIVE RESULTS OF DIFFERENT METHODS IN DIFFERENT CORPORA WITH AROUSAL RATINGS

Arousal		EmoBank (reader)		EmoBank (writer)		FB		CVAT	
		MAE	*r*	MAE	*r*	MAE	*r*	MAE	*r*
Lexicon	Lexicon-*w*AM	0.689	0.326	0.698	0.311	0.985	0.719	0.985	0.268
	Lexicon-*w*GM	0.687	0.325	0.705	0.305	0.981	0.722	0.996	0.263
Regression	Regression-AVR	0.689	0.335	0.692	0.322	0.935	0.758	0.862	0.286
	Regression-MVR	0.687	0.336	0.689	0.328	0.923	0.765	0.842	0.289
Conventional NN	CNN	0.490	0.421	0.498	0.428	0.790	0.843	0.788	0.453
	RNN	0.576	0.375	0.556	0.376	0.894	0.836	0.816	0.290
	LSTM	0.474	0.464	0.486	0.445	0.769	0.869	0.751	0.472
Structured NN	Attention LSTM	0.454	0.480	0.459	0.471	0.694	0.877	0.758	0.489
	Tree-LSTM	0.466	0.478	0.464	0.468	0.690	0.880	0.747	0.504
	CNN-LSTM	0.468	0.477	0.470	0.464	0.707	0.876	0.729	0.523
	Two-layer Attention LSTM (HAN)	0.450	0.484	0.458	0.485	0.629	0.904	0.694	0.551
	Regional CNN-LSTM (Sequential)	0.456	0.482	0.456	0.476	0.644	0.895	0.689	0.557
	Regional CNN-LSTM (Tree)	**0.432**	**0.504**	**0.448**	**0.499**	**0.623**	**0.905**	**0.645**	**0.580**

*Regional CNN-LSTM (Tree) vs. Two-layer Attention LSTM (HAN) differ significantly ($p < 0.05$)

TABLE VII
COMPARISON OF THE REGIONAL STRUCTURE AT DIFFERENT PARSE TREE DEPTHS

Depth	**Region Division on Sentence**
Depth=3	A much more successful translation than its most famous previous film adaptation | , writer-director Anthony Friedman 's | similarly updated 1970 British production .
Depth=4	A much more successful translation | than its most famous previous film adaptation | , writer-director Anthony Friedman 's | similarly | updated | 1970 British production .
Depth=5	A | much more successful | translation | than | its most famous | previous film adaptation | , writer-director Anthony Friedman 's | similarly | updated | 1970 British production .
Depth=3	An interesting look | behind | the scenes of Chicago-based rock group Wilco
Depth=4	An interesting look | behind | the scenes | of Chicago-based rock group Wilco
Depth=5	An interesting look | behind | the scenes | of | Chicago-based rock group Wilco
Depth=3	It 's tough to watch | , but | it 's a fantastic movie
Depth=4	It | 's tough to watch | , but | it | 's a fantastic movie
Depth=5	It | 's | tough to watch | , but | it | 's | a fantastic movie

F. Structured Analysis

To investigate the effect of regional division in the tree-structured regional CNN-LSTM model, we summarized some interesting structures discovered by regional division. As shown in Table VII, *much more successful*, *an interesting look*, *tough to watch* and *a fantastic movie* are important and task-relevant phrases for affective rating prediction. It is also observed that region length divided using parse tree is flexible.

Based on the explicit structure annotation provided by the parser, the regional division strategy tends to discover interesting and task-relevant phrases. Table VIII lists more examples of different types found by regional division. With the predefined structures, the obtained noun and prepositional phrases, are usually expressive, task-relevant, and longer than verb phrases.

TABLE VIII
REGIONAL DIVISION EXAMPLES IN THE STRUCTURED REGIONAL CNN-LSTM MODEL

Type	Examples
Noun Phrase	an equally impressive degree the best company much more successful
Verb Phrase	becoming more interesting lose credibility tough to watch
Prep. Phrase	of a mysterious explorer from silent vents of a historical marker
Special Phrase	as long as his masterpiece hundred thousand fans a dozen times
Attributive Clause	which was colorful which was a bad idea that are struggling to provide
Adverbial Clause	whenever a wildlife crisis emerges until the film is well because it was so endlessly

TABLE IX
STATISTICS OF REGIONAL DIVISION IN THE TEST SET OF EACH CORPUS

Corpus	Tokens Mean	Valence		Arousal	
		Regions	Token per Region	Regions	Token per Region
SST	19.4	4.56	4.86	-	-
EB (reader)	17.6	4.23	4.21	4.23	4.21
EB (writer)	17.6	4.31	4.35	3.98	4.95
FB	19.7	4.48	4.97	4.48	4.97
CVAT	33.1	8.25	4.08	8.25	4.08

In addition to grammatical phrases, the proposed regional division strategy can also detect some special phrases and interesting clauses, including both attributive and adverbial clauses.

In addition, the comparative results in Tables V and VI show that the tree-structured regional CNN-LSTM model outperformed other structured models, indicating that the divided structure may be more task-relevant and advantageous than those obtained by self-attention or parser tree composition.

Table IX presents the statistics of the structures discovered by regional division, including the mean number of regions and tokens per phrase. The mean number of tokens in each region is stable across different datasets, proving the observation that regions with 4 or 5 words allow for the discovery of task-relevant structures and better sentence representations for dimensional sentiment analysis.

V. CONCLUSION

This study presents a tree-structured regional CNN-LSTM model to predict the VA ratings of texts. The proposed model can capture both local (regional) information within sentences and long-distance dependencies across regions. To further investigate the implicit structures, we propose a regional division strategy to identify regions at different depths of a predefined parser tree so that the structural information can be further incorporated to improve prediction performance. Experimental results show that the proposed method outperforms regression-, conventional NN-based and structured methods from previous studies. In addition, the proposed model can discover task-relevant and advantageous regions, which are linguistic grammar blocks, such as phrases and clauses.

Future work will attempt to apply a more intelligent method to discover linguistic regions by using reinforcement learning. We will also generalize the idea of regional division and structure discovery to other tasks and domains.

REFERENCES

[1] B. Pang and L. Lee, "Opinion mining and sentiment analysis," *Found. Trends Inf. Retriev.*, vol. 1, no. 2, pp. 91–231, 2006.

[2] R. A. Calvo, S. Member, S. D. Mello, and I. C. Society, "Affect detection: An interdisciplinary review of models, methods, and their applications," *IEEE Trans. Affect. Comput.*, vol. 1, no. 1, pp. 18–37, Jan. 2010.

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

[4] R. Feldman, "Techniques and applications for sentiment analysis," *Commun. ACM*, vol. 56, no. 4, pp. 82–89, 2013.

[5] S. M. Mohammad, "Sentiment analysis: Detecting valence, emotions, and other affectual states from text," *Emotion Meas.*, 2016, ch. 9, pp. 201–237.

[6] A. Kennedy and D. Inkpen, "Sentiment classification of movie and product reviews using contextual valence shifters," *Comput. Intell.*, vol. 22, no. 2, pp. 110–125, 2006.

[7] G. Mishne and M. De Rijke, "MoodViews: Tools for blog mood analysis," in *Proc. AAAI Spring Symp. Comput. Approaches Analysing Weblogs*, 2006, pp. 153–154.

[8] N. Li and D. Dash, "Using text mining and sentiment analysis for online forums hotspot detection and forecast," *Decis. Support Syst.*, vol. 48, no. 2, pp. 354–368, 2010.

[9] M. De Marne, C. D. Manning, and C. Potts, "Was it good? It was provocative. Learning the meaning of scalar adjectives," in *Proc. 48th Annu. Meeting Assoc. Comput. Linguist.*, 2010, pp. 167–176.

[10] G. Paltoglou, M. Theunis, A. Kappas, and M. Thelwall, "Predicting emotional responses to long informal text," *IEEE Trans. Affect. Comput.*, vol. 4, no. 1, pp. 106–115, Mar. 2013.

[11] G. Paltoglou and M. Thelwall, "Seeing stars of valence and arousal in blog posts," *IEEE Trans. Affect. Comput.*, vol. 4, no. 1, pp. 116–123, Jan.–Mar. 2013.

[12] L.-C. Yu, J.-L. Wu, P.-C. Chang, and H.-S. Chu, "Using a contextual entropy model to expand emotion words and their intensity for the sentiment classification of stock market news," *Knowl. Based Syst.*, vol. 41, pp. 89–97, 2013.

[13] R. A. Calvo and S. Mac Kim, "Emotions in text: Dimensional and categorical models," *Comput. Intell.*, vol. 29, no. 3, pp. 527–543, 2013.

[14] P. Ekman, "An argument for basic emotions," *Cogn. Emotion*, vol. 6, no. 3, pp. 169–200, 1992.

[15] R. Plutchik, *The Emotions*, Lanham, MD, USA: Univ. Press Amer., 1991.

[16] J. A. Russell, "A circumplex model of affect," *J. Pers. Soc. Psychol.*, vol. 39, no. 6, pp. 1161–1178, 1980.

[17] M. M. Bradley and P. J. Lang, "Measuring emotion: The self-assessment manikin and the semantic differential," *J. Behav. Therapy Exp. Psychiatry*, vol. 25, no. 1, pp. 49–59, 1994.

[18] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," in *Proc. Adv. Neural Inf. Process. Syst.*, 2013, pp. 3111–3119.

[19] T. Mikolov, G. Corrado, K. Chen, and J. Dean, "Efficient estimation of word representations in vector space," in *Proc. Int. Conf. Learn. Represent.*, 2013, pp. 1–12.

[20] J. Pennington, R. Socher, and C. D. Manning, "GloVe: Global vectors for word representation," in *Proc. Conf. Empir. Methods Natural Lang. Process.*, 2014, pp. 1532–1543.

[21] L.-C. Yu, J. Wang, K. R. Lai, and X. Zhang, "Refining word embeddings using intensity scores for sentiment analysis," *IEEE/ACM Trans. Audio, Speech, Lang. Process.*, vol. 26, no. 3, pp. 671–681, Mar. 2018.

[22] Y. Kim, "Convolutional neural networks for sentence classification," in *Proc. Int. Conf. Empir. Methods Natural Lang. Process.*, 2014, pp. 1746–1751.

[23] N. Kalchbrenner, E. Grefenstette, and P. Blunsom, "A convolutional neural network for modelling sentences," in *Proc. 52nd Annu. Meeting Assoc. Comput. Linguist.*, 2014, pp. 655–665.

[24] A. Graves, *Supervised Sequence Labelling*. Berlin, Germany: Springer, 2012.

[25] O. Irsoy and C. Cardie, "Opinion mining with deep recurrent neural networks," in *Proc. Conf. Empir. Methods Natural Lang. Process.*, 2014, pp. 720–728.

[26] K. Cho, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, "Learning phrase representations using RNN encoder-decoder for statistical machine translation," in *Proc. Conf. Empir. Methods Natural Lang. Process.*, 2014, pp. 1724–1734.

[27] X. Wang, Y. Liu, C. Sun, B. Wang, and X. Wang, "Predicting polarities of tweets by composing word embeddings with long short-term memory," in *Proc. 53rd Annu. Meeting Assoc. Comput. Linguist.*, 2015, pp. 1343–1353.

[28] P. Liu, S. Joty, and H. Meng, "Fine-grained opinion mining with recurrent neural networks and word embeddings," in *Proc. Conf. Empir. Methods Natural Lang. Process.*, 2015, pp. 1433–1443.

[29] M. Iyyer, V. Manjunatha, J. Boyd-Graber, and H. Daumé III, "Deep unordered composition rivals syntactic methods for text classification," in *Proc. 53rd Annu. Meeting Assoc. Comput. Linguist.*, 2015, pp. 1681–1691.

[30] A. Joulin, E. Grave, P. Bojanowski, and T. Mikolov, "Bag of tricks for efficient text classification," in *Proc. 15th Conf. Eur. Chapter Assoc. Computat. Linguistics*, 2016, pp. 427–431.

[31] P. Bojanowski, E. Grave, A. Joulin, and T. Mikolov, "Enriching word vectors with subword information," *Trans. Assoc. Computat. Linguistics*, vol. 5, pp. 135–146, 2017.

[32] M. M. Bradley and P. J. Lang, "Affective norms for English words (ANEW): instruction manual and affective ratings," Tech. Rep. C-1, Center Res. Psychophysiology, Univ. Florida, Gainesville, FL, USA, 1999.

[33] W.-L. Wei, C.-H. Wu, and J.-C. Lin, "A regression approach to affective rating of Chinese words from ANEW," in *Proc. Int. Conf. Affect. Comput. Intell. Interact.*, 2011, pp. 121–131.

[34] N. Malandrakis, A. Potamianos, E. Iosif, and S. Narayanan, "Kernel models for affective lexicon creation," in *Proc. Annu. Conf. Int. Speech Commun. Assoc., Interspeech*, 2011, pp. 2977–2980.

[35] L.-C. Yu, J. Wang, K. R. Lai, and X. Zhang, "Predicting valence-arousal ratings of words using a weighted graph method," in *Proc. 53rd Annu. Meeting Assoc. Comput. Linguist.*, 2015, pp. 788–793.

[36] N. Malandrakis, A. Potamianos, E. Iosif, and S. Narayanan, "Distributional semantic models for affective text analysis," *IEEE Trans. Audio, Speech Lang. Process.*, vol. 21, no. 11, pp. 2379–2392, Nov. 2013.

[37] J. Wang, L. C. Yu, K. R. Lai, and X. Zhang, "Community-based weighted graph model for valence-arousal prediction of affective words," *IEEE/ACM Trans. Audio, Speech, Lang. Process.*, vol. 24, no. 11, pp. 1957–1968, Nov. 2016.

[38] D. Gökçay, E. Işbilir, and G. Yildirim, "Predicting the sentiment in sentences based on words: An exploratory study on ANEW and ANET," in *Proc. 3rd IEEE Int. Conf. Cogn. Infocommunications*, 2012, pp. 715–718.

[39] S. Lai, L. Xu, K. Liu, and J. Zhao, "Recurrent convolutional neural networks for text classification," in *Proc. 29th AAAI Conf. Artif. Intell.*, 2015, pp. 2267–2273.

[40] K. S. Tai, R. Socher, and C. D. Manning, "Improved semantic representations from tree-structured long short-term memory networks," in *Proc. 53rd Annu. Meeting Assoc. Comput. Linguist.*, 2015, pp. 1556–1566.

[41] M. Huang, Q. Qian, and X. Zhu, "Encoding syntactic knowledge in neural networks for sentiment classification," *ACM Trans. Inf. Syst.*, vol. 35, no. 3, pp. 1–27, 2017.

[42] R. Socher, A. Perelygin, and J. Wu, "Recursive deep models for semantic compositionality over a sentiment treebank," in *Proc. Conf. Empir. Methods Natural Lang. Process.*, 2013, pp. 1631–1642.

[43] Q. Qian *et al.*, "Learning tag embeddings and tag-specific composition functions in recursive neural network," in *Proc. 53rd Annu. Meeting Assoc. Comput. Linguist.*, 2015, pp. 1365–1374.

[44] Z. Lin *et al.*, "A structured self-attentive sentence embedding," in *Proc. Int. Conf. Learn. Represent.*, 2017, pp. 1–15.

[45] Z. Yang, D. Yang, C. Dyer, X. He, A. Smola, and E. Hovy, "Hierarchical attention networks for document classification," in *Proc. 15th Annu. Conf. North Amer. Chapter Assoc. Comput. Linguist., Human Lang. Technol.*, 2016, pp. 1480–1489.

[46] Y. LeCun, L. Bottou, G. B. Orr, and K.-R. Muller, "Efficient backprop," *Neural Netw.: Tricks Trade*, 2nd Ed., Part Lecture Notes, (Comput. Sci., series), vol. 7700, pp. 9–48, 2012.

[47] S. Buechel and U. Hahn, "EmoBank: Studying the impact of annotation perspective and representation format on dimensional emotion analysis," in *Proc. 15th Conf. Eur. Chapter Assoc. Comput. Linguist.*, 2017, pp. 578–585.

[48] S. Buechel and U. Hahn, "Readers vs. writers vs. texts: Coping with different perspectives of text understanding in emotion annotation," in *Proc. 11th Linguist. Annotation Workshop*, 2016, pp. 1–12.

[49] D. Preotiuc-Pietro, H. Schwartz, and G. Park, "Modelling valence and arousal in facebook posts," in *Proc. Workshop Comput. Approaches Subjectivity, Sentiment Social Media Anal.*, 2016, pp. 9–15.

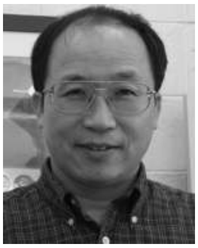
[50] L.-C. Yu *et al.*, "Building Chinese affective resources in valence-arousal dimensions," in *Proc. 15th Annu. Conf. North Amer. Chapter Assoc. Comput. Linguist., Human Lang. Technol.*, 2016, pp. 540–545.

[51] A. Graves, N. Jaitly, and A. R. Mohamed, "Hybrid speech recognition with deep bidirectional LSTM," in *Proc. IEEE Workshop Autom. Speech Recognit. Understand.*, 2013, pp. 273–278.

[52] A. B. Warriner, V. Kuperman, and M. Brysbaert, "Norms of valence, arousal, and dominance for 13,915 English lemmas," *Behav. Res. Methods*, vol. 45, no. 4, pp. 1191–1207, 2013.

Jin Wang received the Ph.D. degree in computer science and engineering from Yuan Ze University, Taoyuan, Taiwan, and in communication and information systems from Yunnan University, Kunming, China. He is an Associate Professor with the School of Information Science and Engineering, Yunnan University, China. His research interests include natural language processing, text mining, and machine learning.

Liang-Chih Yu received the Ph.D. degree in computer science and information engineering from National Cheng Kung University, Tainan, Taiwan. He is a Professor with the Department of Information Management, Yuan Ze University, Taoyuan City, Taiwan. He was a Visiting Scholar with the Natural Language Group, Information Sciences Institute, University of Southern California, from 2007 to 2008, and with DOCOMO Innovations for three months in 2018. His research interests include natural language processing, sentiment analysis, computer-assisted language learning. He is currently Board Member and Convener of SIGCALL of the Association for Computational Linguistics and Chinese Language Processing, and is an Editorial Board Member of *International Journal of Computational Linguistics and Chinese Language Processing*. His team has developed systems that ranked first in IJCNLP 2017 Task 4: Customer Feedback Analysis, and second in the recent SemEval and BEA shared task competitions.



K. Robert Lai received the Ph.D. degree in computer science from North Carolina State University, Raleigh, NC, USA, in 1992. He is a Professor with the Department of Computer Science and Engineering, and the Director of Innovation Center for Big Data and Digital Convergence, Yuan Ze University, Taiwan. His research interests include big data analytics, agent technologies, and mobile computing.

Xuejie Zhang received the Ph.D. degree in computer science and engineering from Chinese University of Hong Kong, Hong Kong, in 1998. He is a Professor with the School of Information Science and Engineering, and the Director of High-Performance Computing Center, Yunnan University, China. His research interests include high performance computing, cloud computing, and big data analytics.