Stacked Residual Recurrent Neural Networks
With Cross-Layer Attention for
Text Classification

YANGYANG LAN\textsuperscript{1,2}, YAZHOU HAO\textsuperscript{1,2}, KUI XIA\textsuperscript{1,2}, BUYUE QIAN\textsuperscript{1,3},
AND CHEN LI\textsuperscript{1,2}, (Member, IEEE)

\textsuperscript{1}School of Computer Science and Technology, Xi’an Jiaotong University, Xi’an 710049, China
\textsuperscript{2}Shaanxi Province Key Laboratory of Satellite and Terrestrial Network Technology R&D, Xi’an Jiaotong University, Xi’an 710049, China
\textsuperscript{3}National Engineering Laboratory for Big Data Analytics, Xi’an Jiaotong University, Xi’an 710049, China

Corresponding author: Chen Li (cli@xjtu.edu.cn)

This work was supported in part by the National Key Research and Development Program of China under Grant 2018YFC0910404, in part by the National Natural Science Foundation of China under Grant 61772409, in part by the Consulting Research Project of the Chinese Academy of Engineering (The Online and Offline Mixed Educational Service System for The Belt and Road Training in MOOC China), in part by the Project of China Knowledge Centre for Engineering Science and Technology, in part by the Innovation Team from the Ministry of Education under Grant IRT_17R86, and in part by the Innovative Research Group of the National Natural Science Foundation of China under Grant 61721002.

\textbf{ABSTRACT} Text classification is a fundamental task in natural language processing and is essential for many tasks like sentiment analysis and question classification etc. As we all know, different NLP tasks require different linguistic features. Tasks such as text classification requires more semantic features than other tasks such as dependency parsing requiring more syntactic features. Most existing methods focus on improving performance by mixing and calibrating features, without distinguishing the types of features and corresponding effects. In this paper, we propose a stacked residual recurrent neural networks with cross-layer attention model to filter more semantic features for text classification, which named SRCLA. Firstly, we build a stacked network structure to filter different types of linguistic features, and then propose a novel cross-layer attention mechanism that exploits higher-level features to supervise the lower-level features to refine the filtering process. Based on this, more semantic features can be selected for text classification. We conduct experiments on eight text classification tasks, including sentiment analysis, question classification and subjectivity classification and compare with a broad range of baselines. Experimental results show that the proposed approaches achieve the state-of-the-art results on 5 out of 8 tasks.

\textbf{INDEX TERMS} Cross-layer attention, stacked residual neural network, bidirectional long short-term memory, feature filtering, text classification.

\section{I. INTRODUCTION}
Text classification is a fundamental task in natural language processing (NLP). It is essential for a number of other tasks, such as sentiment analysis [1], question classification [2], and subjectivity classification [3]. Naturally, it has attracted considerable attention from many researchers, and various types of models have been proposed. A conventional method, e.g. a the bag-of-words (BoW) model, would treat a text as a set of unordered words [4]. In recent years, the deep learning models have been widely used for the task to more extensively incorporate feature information, for example, recursive neural networks [1], [5], convolutional neural networks (CNN) [6], [7], and recurrent neural networks (RNN) [8], [9]. More recently, attention-based models and external knowledge enhanced models become the research hotspots.

However, most existing methods merely improve the performance via combining different features without distinguishing them by types. It would be very useful to investigate the corresponding effects with the different types. As we all know, different NLP tasks require different linguistic features. The tasks like text classification require more semantic features than other tasks such as dependency parsing relying more syntactic features, as shown in Figure 1.

Inspired by the previous works [10]–[12], which have shown that the different layers of stacked model encode different types of information. Specifically, the higher level layers capture more semantic features and the lower level
layers capture more syntactic features, as shown in Figure 2. More specifically, if we assume that each layer of features contains three kinds of features: morphology, syntactic and semantics, the proportion of the three types of features is 1. From the low level to the high level, the proportion of three types of features in each layer will change roughly as shown in the left part of Figure 2. In addition, due to the selectivity of attention mechanism, we propose a novel cross-layer attention mechanism that exploits higher-level features to supervise the lower-level features to filter more semantic features for text classification.

In this paper, we propose a stacked residual model with cross-layer attention method (SRCLA) capable of filtering and selecting features for a specific task. Therefore, we first utilize stacked structure to roughly filter linguistic features, and then propose a cross-layer attention method to refine the filtering process, and finally introduce morphological features, e.g. character-level representation, at the lower level of the model to validate the filtering process. The cross-layer attention method uses information reflexion to supervise low-level linguistic features with high-level semantic features, which can also be referred as the high-to-low attention in the paper. Furthermore, the cross-layer attention method is also able to mitigate the problem of long-distance dependence.

The contributions of this paper can be summarized as the following:

- We propose a stacked residual cross-layer attention model (SRCLA) to filter linguistic features. The model first utilizes the stacked structure to filter the linguistic features. Then, the proposed cross-layer attention is used to refine the filtering process. To the best of our knowledge, we are the first to propose the cross-layer attention method.
- We validate SRCLA on eight classification tasks, including sentiment analysis, question classification and subjectivity classification. The experimental results indicate that our approaches achieve the state-of-the-art results on 5 out of 8 tasks compared with a broad range of baselines.
- We have done detailed experiments to verify the validity and rationality of our model, including the effect of cross-layer attention, the effect of model layers, the convenience and effectiveness of integrating external knowledge, and the visualization of attention weights.

The remainder of the paper is organized as the following. In section 2, the related work about text classification is reviewed. Section 3 presents the proposed model structures for text classification in detail. Section 4 describes the details about the setup of the experiments. Section 5 presents the experimental results and the analyses. The conclusion and the future work are in section 6.

II. RELATED WORK

Deep learning based neural network models have achieved great improvement on text classification tasks. These models generally consist of a projection layer that maps words of text to vectors, and then combines the vectors with different neural networks to make a fixed-length representation. According to the relevance of our work, we divide it in the following categories.

A. RECURRENT NEURAL NETWORKS

RNN has obtained much attention because of the superior ability to preserve the sequence information over time. Some variants (e.g., LSTM [13] and GRU [14]) have also been proposed because RNN can’t handle the gradient vanishing and long-distance dependencies problem. In order to utilize both the past and the future information, BiLSTM [15] are proposed to extend the unidirectional LSTM by introducing a second hidden layer. Reference [8] generalized LSTM to Tree-LSTM where each LSTM unit can gain information from its children units. Reference [9] introduced BiLSTM with attention mechanism to automatically select features that have decisive effect on classification.

B. STACKED NEURAL NETWORKS

Due to the strong representation ability of stacked neural networks, stacked neural networks have been widely used in many tasks. Network depth is of central importance for neural networks as a powerful machine paradigm [16]. Theoretical evidence indicates that deeper networks can be exponentially more efficient at representing certain function classes [17], [18]. Although the strong representation ability of deep model, deep networks suffer from what are commonly referred to as the vanishing and exploding gradient problems [19], [20]. In order to overcome the vanishing and exploding...
gradient problems of deep model, Highway networks [21] and Residual networks [22] are proposed. In recent years, deep networks have been widely used in many NLP tasks, such as text classification [23], language modeling [24], etc. Besides, the Transformer model [25] of using attention only also takes six blocks. Reference [10]–[12] have shown that different layers of deep BiRNNs encode different types of information, specifically that the higher-level layers capture context-dependent aspects of word meaning while the lower-level capture aspects of syntax, as shown in Figure 2. Inspired by this, we propose our model to better use information of different layers.

C. ATTENTION-BASED NEURAL NETWORKS

Attention-based Neural Networks have attracted growing interest due to their ability to explicitly capture the importance of context words. Reference [9] proposed a BiLSTM-Atten model for text classification, which applies attention mechanism in BiLSTM hidden outputs to select the important features for the task. Reference [26] proposed a hierarchical attention model for document classification, which consists of two-level attention mechanisms (word-level and sentence-level). [27] proposed a directional self-attention for RNN/CNN-free natural language understanding. Besides, Attention-based networks and its variants have also been widely used in other tasks, such as machine translation [25], [28], etc.

D. OTHER NEURAL NETWORKS

In addition to the models described above, many other neural networks have also been proposed for text classification. [1], [5] introduced recursive neural tensor network to build representations of phrases and sentences by combining neighbour constituents based on the parsing tree. Reference [6], [7] utilized Convolution neural networks for text classification. Reference [29] attempted to leverage external linguistic knowledge for sentiment classification. Some works based on supervised topic model [30], semantically rich hybrid model [31], noisy label aggregation [32] and other methods are also proposed for text classification.

III. MODEL

Figure 3 is the structure of the SRCLA model, which contains four processes: ①, ②, ③, ④. In this section, We will elaborate the details of each process in the model corresponding to ①, ②, ③, ④ in Figure 3.

A. WORD EMBEDDING

As shown in Figure 3, ① represents the process of word embedding. In a NLP task, a sentence is often treated as a sequence of discrete tokens, i.e. words or characters. We denote a sentence as \( V = [v_1, v_2, \ldots, v_n] \), in which \( v_j \) could be a one-hot vector which dimension length equals to the number of distinct tokens \( N \). A pre-trained token embedding, e.g. Glove [33], is applied to \( V \) and all discrete tokens are transformed to a sequence of low-dimensional dense vector representations \( X = [x_1, x_2, \ldots, x_n] \) with \( x_j \in \mathbb{R}^d \). This pre-process can be written as the equation,

\[
X = W^{(e)} V
\]  

where \( W^{(e)} \in \mathbb{R}^{d \times N} \) denotes word embedding weight matrix, \( X \in \mathbb{R}^{d \times n} \) denotes the low-dimensional dense vector representations of a sentence.

B. STACKED RESIDUAL BILSTM

The specific stacked residual BiLSTM structure in SRCLA model is shown in Figure 4, which corresponding to the process ② in Figure 3. LSTM was firstly proposed by [13]
to overcome gradient vanishing problem of RNN. The main idea is to introduce an adaptive gating mechanism, which decides the degree to keep the previous state and memorize extracted features of the current data input. The stacked BiLSTM is illustrated in ② in Figure 3. Given a sentence \( X = [x_1, x_2, \ldots, x_n] \), where \( n \) is the length of the input sentence. \( x_i \in \mathbb{R}^d \) is the \( d \)-dimensional word vector corresponding to the \( i \)-th word in the sentence \( X \). LSTM processes it word by word, at time-step \( t \), the memory \( c_t \) and the hidden state \( h_t \) are updated with the following equations:

\[
\begin{bmatrix}
i_t \\ f_t \\ o_t \\ c_t
\end{bmatrix} = \begin{bmatrix}
\sigma & \sigma & \sigma \tanh
\end{bmatrix} \begin{bmatrix}
i_{t-1, x_t} \\
0
\end{bmatrix}
\]

(2)

\[
c_t = f_t \odot c_{t-1} + i_t \odot \tilde{c}_t
\]

(3)

\[
h_t = o_t \odot \tanh(c_t)
\]

(4)

Here, \( h_t \in \mathbb{R}^m \), in which \( m \) denotes the hidden dimension of LSTM. \( x_t \) is the input at the current time-step. \( i, f \) and \( o \) are the input gate activation, the forget gate activation and the output gate activation respectively. \( \tilde{c} \) is the current cell state. \( \sigma \) denotes the logistic sigmoid function and \( \odot \) denotes element-wise multiplication.

For the sequence modeling tasks, it is beneficial to have access to the past context as well as the future context. In this paper, BiLSTM is utilized to capture past and future information. So, the output of the \( i \)-th word is shown in the following equation:

\[
h_t = \overrightarrow{h_t} \oplus \overleftarrow{h_t}
\]

(5)

Here, \( \oplus \) denotes the concatenation of hidden states of the forward LSTM and the backward LSTM. So, \( h_t \in \mathbb{R}^{2m} \).

In order to prevent the gradient vanishing in the stacked model, we introduce a residual connection between any two adjacent layers. The specific structure of stacked residual BiLSTM is shown in Figure 4. As shown in Figure 4, the concatenation of the hidden state \( h_t^j \) of the \( j \)-th BiLSTM unit and the hidden state \( h_{t-1}^j \) of the \( (l-1) \)th BiLSTM unit is used as the input to the \( (l+1) \)th BiLSTM unit at the time-step \( t \), which is shown in the following equation:

\[
h_t^{l+1} = \text{BiLSTM}(h_{t-1}^j, [h_t^j \oplus h_{t-1}^j])
\]

(6)

Here, \( \oplus \) denotes the concatenation operation.

### C. CROSS-LAYER ATTENTION

As shown in Figure 3, ③ denotes the process of the Cross-Layer Attention. Let the number of BiLSTM be \( L \). The input sentence \( X = [x_1, x_2, \ldots, x_n] \), where \( X \in \mathbb{R}^{n \times d} \). \( n \) denotes the length of the sentence and \( d \) denotes the dimension of word embedding. The hidden output of the \( K \)th BiLSTM \( H^K = [h^K_1, h^K_2, \ldots, h^K_n] \), where \( H^K \in \mathbb{R}^{n \times 2d^K} \), \( d^K \) denotes the hidden dimension of the \( K \)th LSTM. We use the row-max-pooling of hidden output \( H^L = [h^L_1, h^L_2, \ldots, h^L_n] \) of \( L \)th BiLSTM as query, as shown in the following equation:

\[
q = h^L_s = \text{MaxPooling}_{row}(H^L)
\]

(7)

where \( q \in \mathbb{R}^{2d^K} \), \( d^L \) denotes the hidden dimension of the \( L \)th LSTM. We copy \( q \) \( n \) times to get query matrix \( Q \), where \( Q \in \mathbb{R}^{n \times 2d^K} \).

Then, we compute the attention weight by the additive method [28], as shown in the following equations:

\[
a = w^T \tanh(W^{(1)}Q + W^{(2)}H^K)
\]

(8)

\[
p(z|H^K, Q) = \text{softmax}(a)
\]

(9)

Here, we take \( \tanh \) function as the activation function. \( p(z = i|H^K, Q) \) denotes the importance of the \( i \)th hidden output of the \( K \)th BiLSTM.

According to the normalized attention weight \( p(z|H^K, Q) \), we get the weighted average representation of the hidden output of the \( K \)th BiLSTM, as shown in the following equation:

\[
h^K_s = \sum_{i=1}^n p(z = i|H^K, Q)h^K_i
\]

(10)

where \( h^K_s \in \mathbb{R}^{2d^K} \), \( d^K \) denotes the hidden dimension of the \( k \)th LSTM. We concatenate \( h^K_s \) and \( h^K_s \) to get the final sentence representation \( h_s \), as shown in the following equation:

\[
h_s = [h^K_s \oplus h^K_s]
\]

(11)

where \( h_s \in \mathbb{R}^{2d^K+2d^L} \). The value of \( K \) depends on the value of \( L \). \( K \) is a subset of all sets made up of elements from \( \{1, 2, \ldots, L-1\} \). Considering the semantic features contained in each layer, the amount of dataset, the running time and the parameters, we specify the values of \( L \) and \( K \), as shown in Table 2. Later, we will conduct a detailed experiment to analyze why we set the values of \( L \) and \( K \) in this way.

### D. MLP CLASSIFIER

For text classification, we take a two-layer perceptron and softmax as the classifier to predict the label \( \hat{y} \) from a discrete set of classes \( Y \). The classifier takes the final sentence
representation \( h_i \) as input and takes \( \text{relu} \) as the activation function, as shown in the following equations:

\[
\begin{align*}
    v_1 &= \text{relu}(W^{(3)}h_i + b^{(3)}) \\
    v_2 &= \text{relu}(W^{(4)}v_1 + b^{(4)}) \\
    \hat{p}(y|v_2) &= \text{softmax}(W^{(5)}v_2 + b^{(5)}) \\
    \hat{y} &= \arg\max_{y_i} \hat{p}(y|v_2)
\end{align*}
\]

We take the cross-entropy loss with L2 regularization as the training objective, as shown in the following equation:

\[
J(\theta) = -\frac{1}{m} \sum_{i=1}^{m} t_i \log(p(y_i)) + \lambda \|\theta\|^2_2
\]

where \( t \in R^m \) is the one-hot represented ground truth. \( p(y) \in R^m \) is the estimated probability for each class by softmax. \( m \) is the number of target classes. \( \lambda \) is an L2 regularization hyper-parameter. \( \theta \) is the model parameters.

### IV. EXPERIMENTAL SETUP

#### A. DATASETS

As we all know, text classification is a fundamental task for a number of more advanced NLP tasks, such as sentiment analysis, problem classification, subjective recognition, etc. In order to test the validity and robustness of our model, eight datasets are carefully selected from many as being representative. The statistics of the datasets are in Table 1.

- **SST-1**: Stanford Sentiment Treebank. It is an extension of MR with train/dev/test splits provided and fine-grained labels (very positive, positive, neutral, negative, very negative), re-labeled by [1].
- **SST-2**: It is the same as SST-1 with neutral reviews removed and binary labels. For both experiments, phrases and sentences are used to train the model, but only the results on the sentences are marked while testing [1].
- **TREC**: TREC question dataset. The task involves classifying a question into one of six question types (whether the question is about person, location, numeric information, etc.) [2].
- **MR**: Movie reviews with one sentence per review. Classification involves detecting positive/negative reviews [34].

#### B. HYPER-PARAMETER SETTINGS AND TRAINING

For the datasets without a development set, we randomly select 10% of the training data as the development set. In the experiments, all word vectors are initialized by Glove 840B pre-trained vectors [33]. The out-of-vocabulary word embedding is initialized by randomly sampling from uniform distribution in \([-0.25, 0.25]\). The word embedding is fine-tuned during training to improve the classification performance. We minimize objective functions by Adagrad [41] with batch size 25 and initial learning rate 0.003. In order to prevent overfitting, we adopt early stop with the tolerance of 5.

For regularization, we use L2 penalty with coefficient \( 10^{-5} \) over the parameters, besides, we also employ Dropout operation [42] with dropout rate of 0.5 for the word embedding, character embedding, the BiLSTM layer and the penultimate layer. Considering the running time and parameters, we specify the values of \( L \) and \( K \), the values are shown in Table 2. The set of LSTM hidden dimension is \([512, 1024, 2048, 4096]\). All weight matrices are initialized by Glorot Initialization [43]. All models are implemented with Pytorch and run on single Nvidia GTX 2080Ti graphic card.

### V. RESULTS AND ANALYSIS

#### A. OVERALL PERFORMANCE

This work implements two models, Stacked-BiLSTM and SRCLA. Table 3 presents the performance of the two models along with the state-of-the-art models on seven classification tasks. Our models achieve the excellent performances on seven classification tasks, especially SRCLA achieves the state-of-the-art results on SST-1, SST-2, MR and CR dataset respectively. According to the results in Table 3, we have the following observations:

- **Subj**: Subjectivity dataset. The task is to classify a sentence as being subjective or objective [3].
- **CR**: Customer reviews of various products (cameras, MP3s etc.). The task is to predict positive/negative reviews [35].
- **MPQA**: Opinion polarity detection. It is a subtask of the MPQA dataset [36].
- **IMDB**: Document-level movie reviews. The classification involves detecting positive/negative reviews [37].

#### B. RESULTS AND ANALYSIS

| Dataset | C | L | Train | Dev | Test | V | \( P_{pre} \) |
|---------|---|---|-------|-----|------|---|------------|
| SST-1   | 5 | 18| 8544  | 1101| 2210 | 16188| 15756      |
| SST-2   | 2 | 19| 6920  | 872 | 1821 | 17836| 17328      |
| TREC    | 6 | 10| 5452  | -   | -    | 500 | 9593 | 9482       |
| MR      | 2 | 21| 10662 | -   | -    | CV  | 18777| 17635      |
| Suby    | 2 | 22| 10000 | -   | -    | CV  | 21335| 19937      |
| CR      | 2 | 19| 3775  | -   | -    | CV  | 5342 | 5297       |
| MPQA    | 2 | 3 | 10066 | -   | -    | CV  | 6246 | 6191       |
| IMDB    | 2 | 258| 25000 | -   | -    | 25000| 112540| 78554      |

#### Table 2. The value of parameter L and its corresponding parameter K.

| L  | K  |
|----|----|
| 2  | 1  |
| 3  | 2  |
| 4  | (3, 2) |
| 5  | (4, 3) |

VOLUME 8, 2020  70405
TABLE 3. Classification results on several standard benchmarks. RNTN: Recursive deep models for semantic compositionality over a sentiment treebank [1]. DRNN: Deep recursive neural networks for compositional language modeling [5]. DCNN: A convolutional neural network for modeling sentences [7]. CNN-nonstatic/MC: Convolutional neural networks for sentence classification [6]. TBCNN: Discriminative neural network model for tree-based convolution [38]. BiLSTM/Tree-LSTM: Improved semantic representations from tree-structured and long-term memory networks [8]. BiLSTM-Att/2DCNN: Text classification improved by integrating Bidirectional LSTM with two-dimensional Max Pooling [9]. DisSAN: DisSAN: Directional Self-Attention Network for RNN/CNN-Free Language Understanding [27]. AdaSent: Self-adaptive hierarchical sentence model [39]. NSCL: Context-Sensitive Lexicon Features for Neural Sentiment Analysis [40]. LR-BiLSTM: Linguistically Regularized LSTM for Sentiment Classification [29]. The best result for each dataset is in bold. The result marked with # are retrieved from [29]. ∗ indicates that our model is significantly better than most baseline models such as LSTM, BiLSTM, Tree-LSTM, LSTM-Att, DisSAN, etc with p < 0.05 based on one-tailed unpaired t-test.

| Model     | Dataset | LSTM   | MR   | Subj | CR   | MPQA |
|-----------|---------|--------|------|------|------|------|
| RNN       | SST-1   | 45.7   | 85.4 | -    | -    | -    |
|           | SST-2   | 49.8   | 86.6 | -    | -    | -    |
| CNN       | DCNN    | 48.5   | 86.8 | 83.0 | -    | -    |
|           | CNN-non-static | 48.0 | 87.2 | 93.6 | 81.5 | 93.4 | 84.3 | 89.5 |
|           | CNN-MC  | 47.4   | 88.1 | 92.2 | 81.1 | 93.2 | 85.0 | 89.4 |
|           | TBCNN   | 51.4   | 87.9 | 96.0 | -    | -    | -    | -    |
| RNN       | BiLSTM  | 46.4   | 84.9 | -    | 77.3#| -    | -    | -    |
|           | Tree-LSTM | 49.1  | 87.5 | -    | 79.3#| -    | -    | -    |
| Attention | BiLSTM-Att | 51.0  | 88.0 | -    | 80.7#| -    | -    | -    |
| Other     | DisSAN  | 49.8   | 82.2 | 93.8 | 81.5 | 93.5 | -    | -    |
|           | AdaSent | 51.72  | 94.2 | 94.2 | 84.8 | 90.1 | -    | -    |
| Our       | BiLSTM-2DCNN | 52.4  | 89.5 | 96.1 | 82.3 | 94.0 | -    | -    |
|           | NSCL    | 51.1   | 89.2 | -    | 82.9 | -    | -    | -    |
|           | LR-BiLSTM | 50.6  | 89.2 | -    | 82.1 | -    | -    | -    |
| Our       | BiLSTM-CLA | 51.3  | 89.0 | 92.6 | 82.5 | 93.8 | 85.2 | 89.6 |
|           | SRCLA   | 52.8*  | 90.2* | 94.2 | 84.0* | 95.0* | 87.8* | 90.5 |

Subj dataset. Compared with BiLSTM, the Stacked-BiLSTM improves the result by a larger margin, which reflects that stacked structure can filter linguistic features. Compared with Stacked-BiLSTM, SRCLA achieves about 1.5% improvement over all datasets, which proves that the cross-layer attention method can refine the filtering process and select more semantic features. In summary, our model is able to filter linguistic features and select more semantic features for text classification.

Secondly, SRCLA can mitigate the long-distance dependence problem. Compared with DiSAN model that adopts the directional self-attention method making the distance between any two words one, our models outperform it on all datasets. This demonstrates the capability on the long-distance dependence problem. To further validate the observation, we apply SRCLA model to the document-level dataset IMDB. The results are in Table 4. As shown in Table 4, compared with a broad range of baselines, SRCLA model achieves the state-of-the-art results by a larger margin, which further prove the observation.

Thirdly, SRCLA can further improve the results by integrating external knowledge conveniently. The previous works, such as RNTN/TBCNN/Tree-LSTM/LR-BiLSTM, have used external knowledge. Our current models outperform them without introducing external knowledge. We can further improve the result by integrating morphological knowledge (e.g. character-level embedding) at the lower layer, introducing syntactic knowledge (e.g. parsing) in middle layer and introducing semantic knowledge (e.g. sentiment lexicon) in higher layer. In the following subsection, we will do further experiments to verify the observation.

Lastly, compared with context feature, n-gram feature is more important in short text classification. Compared the BiLSTM-2DCNN model that capture more n-gram feature by CNN model, we don’t get the best result on TREC dataset. Compared the AdaSent model that captures more n-gram feature by complex mathematical methods, we don’t get the best result on MPQA dataset. We think that the sentence length of TREC and MPQA dataset is relatively short, so n-gram feature may play more important role than context feature in short text classification.

B. EFFECTIVENESS OF CROSS-LAYER ATTENTION

In order to analyse the effectiveness of cross-layer attention, we conduct the ablation experiments on MR, Subj, CR and MPQA dataset. The experimental results are shown in Table 5. We have the following observations:

Firstly, the cross-layer attention is effective. Compared with SRCLA, SRCLA-CLA decreases 1.4% on average over all dataset, which demonstrates the observation on the whole. Compared with SRCLA, SRCLA-CLA+ decreases 1.1% on BiLSTM-2DCNN model that capture more n-gram feature by CNN model, we don’t get the best result on TREC dataset. Compared the AdaSent model that captures more n-gram feature by complex mathematical methods, we don’t get the best result on MPQA dataset. We think that the sentence length of TREC and MPQA dataset is relatively short, so n-gram feature may play more important role than context feature in short text classification.

TABLE 4. Classification results about IMDB dataset on several standard benchmarks. LSTM+LA: Neural sentiment classification with user and product attention [44]. LSTM+CBA+LA: A cognition based attention model for sentiment analysis [45]. LSTM+dynamic skip: Long Short-Term Memory with Dynamic Skip Connections [46]. The result marked with # are retrieved from [46]. ∗ indicates that our model is significantly better than LSTM, LSTM+LA, LSTM+CBA+LA, LSTM-Att, LSTM+dynamic skip with p < 0.05 based on one-tailed unpaired t-test.

| Model     | IMDB   |
|-----------|--------|
| LSTM      | 89.1#  |
| LSTM + LA | 89.3   |
| LSTM + CBA + LA | 90.1 |
| LSTM-Att  | 89.4#  |
| LSTM + dynamic skip | 90.1 |
| Stacked-BiLSTM | 90.8 |
| SRCLA     | 92.3*  |
average over all datasets by concatenating directly the representation of previous layer and the representation of last layer. As shown in Table 6, in all datasets, when \( L = 3 \) and \( K = \{2\} \), Subj dataset achieves the best result when \( L = 2, K = \{1\} \), and MPQA dataset achieves the best result when \( L = 3, K = \{1, 2\} \). It is supposed to be related with the size of datasets, the number of model parameters and the linguistic features required for tasks.

Lastly, how to determine the optimal combination of \( L \) and \( K \). As shown in Table 6, different values of \( L \) have great influence on the results, and when the value of \( L \) is determined, the value of \( K \) has little influence on the results. So, we suggest that we first use \( k = \lceil (L - 1) \rceil \) to get the optimal value of \( L \), then use \( K = \lceil (L - 1, L - 2) \ldots \) to obtain the optimal value of \( K \), and we suggest that the number of elements in \( K \) should meet the following formula:

\[
|K| \leq \left\lceil \frac{L}{2} \right\rceil
\]

Here, \(|K|\) represents the number of elements in \( K \). \( \lceil \rceil \) indicates rounding up.

**D. CONVENIENCE AND EFFECTIVENESS OF INTRODUCING EXTERNAL KNOWLEDGE**

In order to verify the convenience and effectiveness of introducing external knowledge into our model, we introduce the morphological knowledge (e.g. character-level embedding) at the lower layer of SRCLA model, which we call Char-CNN-SRCLA. Specifically, we apply CNN model to character embedding, take the output of CNN as character representation, and use the concatenation of character-level representation and corresponding word embedding as the input of SRCLA model. Here, the character embedding is initialized by randomly sampling from uniform distribution in \([-0.25, 0.25]\), the dimension of character embedding is 50. The character embedding is fine-tuned during training to improve the classification performance. We set CNN filters of width \([3, 4, 5]\) of size \([50, 50, 50]\), so the dimension of character representation is 150. The experiments are conducted on MR, Subj, CR, MPQA datasets. The experimental results are shown in Table 7. According to Table 7, we have the following observations:

**TABLE 5.** Effectiveness of Cross-layer attention. SRCLA-CLA: Without the cross-layer attention, and is actually a stacked residual model (the representation of \( L^{th} \) layer as the final presentation). SRCLA-CLA+: Based on SRCLA-CLA, concatenate directly the representation of \( K^{th} \) layer and the representation of \( L^{th} \) layer as the final representation. The best result is in bold. * indicates that our model is significantly better than Stacked-BiLSTM, SRCLA-CLA with \( p < 0.05 \) based on one-tailed unpaired t-test.

| Model          | MR     | Subj    | CR     | MPQA   |
|----------------|--------|---------|--------|--------|
| Stacked-BiLSTM | 82.5   | 93.3    | 85.2   | 89.6   |
| SRCLA          | 84.0*  | 95.0*   | 87.8*  | 90.5*  |
| SRCLA-CLA     | 82.7   | 93.9    | 85.5   | 89.8   |
| SRCLA-CLA+    | 83.0   | 94.1    | 86.0   | 90.0   |

**TABLE 6.** The classification result of MR, Subj, MPQA and CR dataset under different \( L \) and \( K \). The best result for each dataset is in bold.

| \( L \) | \( K \) | MR   | Subj | CR   | MPQA |
|--------|--------|------|------|------|------|
| 2      | \( \{1\} \) | 83.3 | 95.0 | 87.2 | 90.0 |
| 2      | \( \{2\} \) | 84.0 | 97.4 | 87.8 | 90.3 |
| 2      | \( \{1, 2\} \) | 83.5 | 94.8 | 87.5 | 90.5 |
| 3      | \( \{1\} \) | 83.6 | 94.7 | 87.5 | 90.2 |
| 3      | \( \{2\} \) | 82.9 | 94.4 | 86.8 | 90.1 |
| 3      | \( \{3\} \) | 83.6 | 94.5 | 87.2 | 90.1 |
| 3      | \( \{1, 2\} \) | 82.9 | 94.6 | 86.9 | 90.2 |
| 3      | \( \{1, 3\} \) | 83.1 | 94.4 | 86.6 | 90.1 |
| 4      | \( \{2, 3\} \) | 83.2 | 94.5 | 86.9 | 90.1 |
| 4      | \( \{1, 2, 3\} \) | 82.9 | 94.4 | 86.5 | 90.1 |

**TABLE 7.** The classification result on MR, Subj, CR and MPQA dataset. Char-CNN-SRCLA: further introduces morphological features at the lower layer of SRCLA. Specifically, concatenate the character-level representation captured by CNN model and word embedding at the input layer of SRCLA. The best result is in bold.

| Model          | MR     | Subj    | CR     | MPQA   |
|----------------|--------|---------|--------|--------|
| Stacked-BiLSTM | 82.5   | 93.8    | 85.2   | 89.6   |
| SRCLA          | 84.0   | 95.0    | 87.8   | 90.5   |
| Char-CNN-SRCLA | 84.2   | 95.3    | 88.1   | 90.9   |
Firstly, introducing morphological knowledge into SRCLA is effective. Compared with SRCLA, Char-CNN-SRCLA increases 0.3% on average over all datasets, which demonstrates the observation.

Secondly, introducing external knowledge into SRCLA is intuitive and convenient. As shown in Figure 2, different layers contain different types of information, so we can introduce corresponding types of external knowledge into corresponding layers.

### E. VISUALIZATION OF ATTENTION

In order to get a better understanding of our model and validate that this model is able to filter and select more semantic features, we visualize the obtained attention weight \( p(z | H^K, Q) \) in Equation (9). Figure 5 shows the attention visualizations for some sentences sampling from MR dataset. The color depth indicates the importance degree of attention weight for the sentence, the darker the more important.

MR belongs to sentiment classification dataset, so we will analyze the results from the perspective of sentiment classification. As we all know, analyzing the emotion of a text is to recognize the emotion expression in the text. According to the habit of daily expression, the emotional expression in the text can be roughly divided into two categories: explicit emotional expression and implicit emotional expression. The explicit emotional expression can be divided into one expression, multiple expressions with the same polarity, multiple expressions with opposite polarity, as shown in the Figure 5. Next, we will analyze the results of four samples one by one.

For the first sample, it belongs to explicit emotional expression and contains only one expression, which is easy to identify for most models. As shown in the Figure 5, our model can select positive emotion words or phrases such as “enjoyable” and “feel good”, which proves the validity of our model.

For the second sample, it belongs to explicit emotional expression and contains multiple expressions, but the polarity is opposite, which is difficult to identify for such samples. As shown in the Figure 5, our model can accurately identify the negative affective phrase “extremely silly peace”, which will not be affected by the positive affective phrases such as “fast packed”, “glitzy”, etc., which proves the robustness of our model.

For the third example, it belongs to implicit emotional expression, which is the most difficult to identify. Because there is no clear emotional expression, it is necessary to distinguish the deep semantics of the text, and the requirements for the model are high. As shown in the Figure 5, our model can select a relatively negative emotional phrase “get ready to take off”, which further verifies that our model can filter and select more semantic features.

### VI. CONCLUSION

In this paper, two models for text classification are presented, which are SRCLA and Char-CNN-SRCLA. They are different from the previous methods that improve performance by mixed linguistic features. The novel models are able to filter linguistic features and select more semantic features for text classification. SRCLA first utilizes a stacked structure to filter linguistic features, and then uses a novel attention method. The method is a cross-layer attention capable of refining the filtering process. The Char-CNN-SRCLA further introduces morphological features at the lower layer of SRCLA to validate the filtering process. The experiments are conducted on eight text classification tasks, and the results show that both SRCLA and Char-CNN-SRCLA achieve excellent performance compared with a broad range of baselines. Especially, the SRCLA and Char-CNN-SRCLA achieve state-of-the-art results on 5 out of 8 tasks. To better understand the effectiveness of the proposed models, a series of elimination experiments are carried out and analysed in the paper. It is demonstrated that filtering linguistic features enhances performance in text classification and possibly in other NLP tasks.

In the future, we would continue the study in three directions. Firstly, we will experiment the different directions of attention. The current direction of SRCLA is from high to

---

**Figure 5.** The attention weight of 2th layer on four samples that are selected from MR dataset. We set \( L = 3, k = \{2\} \) on MR dataset. \( P \): indicates that emotional labels are positive. \( N \): indicates that emotional labels are negative.
low, which can select more semantic features. Could a low- to-high attention assist in selecting features for the more syntactic-related tasks like part-of-speech tagging? Secondly, we would integrate different external knowledge at the different layer (e.g., Char-CNN-SRLCA). Last but not least, we will apply our model to other NLP tasks, such as natural language inference, aspect-level sentiment analysis, etc.

ACKNOWLEDGMENT

The authors thank to Prof. Chen Li for his guidance and assistance and Yazhou Hao, Kui Xia, and Buye Qian for their help.

REFERENCES

[1] R. Socher, A. Perelygin, J. Wu, J. Chuang, C. D. Manning, A. Y. Ng, and C. PottsProc., “Recursive deep models for semantic compositionality over a sentiment treebank,” in Proc. EMNLP, 2013, pp. 1631–1642.

[2] X. Li and D. Roth, “Learning question classifiers,” in Proc. 19th Int. Conf. Comput. Linguistics, 2002, pp. 1–7.

[3] B. Pang and L. Lee, “A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts,” in Proc. 42nd Annu. Meeting Assoc. Comput. Linguistics (ACL), 2004, pp. 271–278.

[4] S. I. Wang and C. D. Manning, “Baselines and bigrams: Simple, good sentiment and topic classification,” in ACL, vol. 2012, pp. 90–94.

[5] O. Iryso and C. Cardie, “Deep recursive neural networks for compositionality in language,” in Proc. NIPS, 2014, pp. 2096–2104.

[6] Y. Kim, “Convolutional neural networks for sentence classification,” in Proc. Conf. Empirical Methods Natural Lang. Process. (EMNLP), 2014, pp. 1746–1751.

[7] N. Kalchbrenner, E. Grefenstette, and P. Blunsom, “A convolutional neural network for modelling sentences,” in Proc. 52nd Annu. Meeting Assoc. Comput. Linguistics, vol. 1, 2014, pp. 655–664.

[8] 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. Linguistics, 7th Int. Joint Conf. Natural Lang. Process., vol. 1, 2015, pp. 1556–1566.

[9] P. Zhou, Z. Qi, S. Zheng, J. Xu, H. Bao, and B. Xu, “Text classification improved by integrating bidirectional LSTM with two-dimensional max pooling,” in Proc. COINLG, 2016, pp. 3485–3495.

[10] M. Peters, M. Neumann, I. Iyyer, M. Gardner, C. Clark, K. Lee, and L. Zettlemoyer, “Deep contextualized word representations,” in Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics, Human Lang. Technol., 2018, pp. 1–15.

[11] K. Hashimoto, C. Xiong, Y. Tsuruoka, and R. Socher, “A joint many-task model: Growing a neural network for multiple NLP tasks,” in Proc. Conf. Empirical Methods Natural Lang. Process., 2017, pp. 1923–1933.

[12] A. Søgaard and Y. Goldberg, “Deep multi-task learning with low level attention assist in selecting features for the more semantic features. Could a low- to-high attention classification,” in Proc. Conf. Empirical Methods Natural Lang. Process., 2015, pp. 649–657.

[13] Y. Kim, J. Yernite, D. Sontag, and A. M. Rush, “Character-aware neural language models,” CoRR, vol. abs/1508.06615, Aug. 2015.

[14] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, “Attention is all you need,” in Proc. NIPS, 2017, pp. 5998–6008.

[15] 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., 2016, pp. 1480–1489.

[16] T. Shen, T. Zhou, G. Long, J. Jiang, S. Pan, and C. Zhang, “Disan: Directional self-attention network for rnn/cnn-free language understanding,” in Proc. AAAI, 2018, pp. 5446–5455.

[17] D. Bahdanau, K. Cho, and Y. Bengio, “Neural machine translation by jointly learning to align and translate,” in Proc. ICLR, 2015, pp. 1–15.

[18] Q. Qian, M. Huang, J. Lei, and X. Zhu, “Linguistically regularized LSTM for sentiment classification,” in Proc. 55th Annu. Meeting Assoc. Comput. Linguistics, vol. 1, 2017, pp. 1–11.

[19] M. Huang, Y. Rao, Y. Liu, H. Xie, and F. L. Wang, “Siamese network-based supervised topic modeling,” in Proc. Conf. Empirical Methods Natural Lang. Process., 2018, pp. 4652–4662.

[20] X. Li, Y. Rao, H. Xie, R. Y. K. Lau, J. Yin, and F. L. Wang, “Bootstrapping social emotion classification with semantically rich hybrid neural networks,” IEEE Trans. Affect. Comput., vol. 8, no. 4, pp. 428–442, Oct. 2017.

[21] X. Zhan, Y. Wang, Y. Rao, H. Xie, Q. Li, F. L. Wang, and T.-L. Wong, “A network framework for noisy label aggregation in social media,” in Proc. 55th Annu. Meeting Assoc. Comput. Linguistics, vol. 2, 2017, pp. 484–490.

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

[23] B. Pang and L. Lee, “Seeing stars: Exploring class relationships for sentiment categorization with respect to rating scales,” in Proc. 43rd Annu. Meeting Assoc. Comput. Linguistics (ACL), 2005, pp. 115–124.

[24] M. Hu and B. Liu, “Mining and summarizing customer reviews,” in Proc. ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining (KDD), 2004, pp. 168–177.

[25] J. Wiebe, T. Wilson, and C. Cardie, “Annotating expressions of opinions and emotions in language,” Lang. Resour. Eval., vol. 39, nos. 2–3, pp. 165–210, May 2005.

[26] A. L. Maas, R. E. Daly, P. T. Pham, D. Huang, A. Y. Ng, and C. Potts, “Learning word vectors for sentiment analysis,” in Proc. ACL, 2011, pp. 142–150.

[27] L. Mou, H. Peng, G. Li, Y. Xu, L. Zhang, and Z. Jin, “Discriminative neural sentence modeling by tree-based convolution,” in Proc. Conf. Empirical Methods Natural Lang. Process., 2015, pp. 2315–2325.

[28] H. Zhao, Z. Lu, and P. Pourpart, “Self-adaptive hierarchical sentence model,” in Proc. IJCAI, 2015, pp. 4069–4076.

[29] Z. Teng, D. T. Vo, and Y. Zhang, “Context-sensitive lexicon features for neural sentiment analysis,” in Proc. Conf. Empirical Methods Natural Lang. Process., 2016, pp. 1629–1638.

[30] H. E. Duché John and S. Yoran, “Adaptive subgradient methods for online learning and stochastic optimization,” J. Mach. Learn. Res., vol. 12, pp. 2121–2159, Jul. 2011.

[31] G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, “Improving neural networks by preventing co-adaptation of feature detectors,” CoRR, vol. abs/1207.0580, Jul. 2012.

[32] X. Glorot and Y. Bengio, “Understanding the difficulty of training deep feedforward neural networks,” in Proc. AISTATS, 2010, pp. 249–256.

[33] H. Chen, M. Sun, C. Tu, Y. Lin, and Z. Liu, “Neural sentiment classification with user and product attention,” in Proc. Conf. Empirical Methods Natural Lang. Process., 2016, pp. 1650–1659.

[34] Y. Long, L. Qin, R. Xiang, M. Li, and C.-R. Huang, “A cognition based attention model for sentiment analysis,” in Proc. Conf. Empirical Methods Natural Lang. Process., 2017, pp. 462–471.

[35] T. Gu, Q. Zhang, L. Zhou, Y. Lin, M. Peng, J. Gong, and X. Huang, “Long short-term memory with dynamic skip connections,” in Proc. AAAI Conf. Artif. Intell., Jul. 2019, pp. 6481–6488.
YANGYANG LAN received the bachelor’s degree in software engineering from Northwestern Polytechnic University, in 2017. He is currently pursuing degree with the Department of Computer Science and Technology, Xi’an Jiaotong University. His research interests include representation learning, sentiment analysis, and natural language processing.

YAZHOU HAO received the bachelor’s degree in computer science and technology from Chang’an University, in 2013. He is currently pursuing the Ph.D. degree with the Department of Computer Science and Technology, Xi’an Jiaotong University. His research interests include representation learning, sentiment analysis, and natural language processing.

KUI XIA received the bachelor’s degree in computer science and technology from Xidian University, in 2018. He is currently pursuing the degree with the Department of Computer Science and Technology, Xi’an Jiaotong University. His research interests include representation learning, sentiment analysis, and natural language processing.

BUYUE QIAN received the Ph.D. degree from the University of California, Davis, in 2013. He was a Former Researcher at the IBM Watson Research Center. He is currently an Associate Professor with the Department of Computer Science and Technology, Xi’an Jiaotong University. In the past few years, he has participated in or led many large research projects, including NSF, onr, NIH, Yahoo, and IBM. He have been published more than 20 articles (CCF class A or B) in several top conferences and journals of data mining and artificial intelligence. In terms of invention patents, there are 23 U.S. patents, involving algorithms, finance, e-commerce, medical, and other fields, of which more than half are the first patent inventors. He has a wide range of interests in data mining and machine learning, specializing in active learning, machine learning-based sorting, spectral clustering, semi supervised learning, and big data analysis-based on hash algorithm. In terms of academic review, he is currently a member of the procedure Committee of several international top academic conferences, including ACM KDD, IJCAI, IEEE ICDM, Siam SDM, ACM CIKM, and ECML/PKDD. He received the awards include the Siam Data Mining Best Research Paper Silver Award, the Yahoo! Research Award, and the IBM Outstanding Research Award. At the same time, he is also a Reviewer of several top international publications, mainly including AI journal, DMKD, the IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING, ACM TKDD, the IEEE TRANSACTIONS ON MULTIMEDIA, ACM TIST, the IEEE TRANSACTIONS ON IMAGE PROCESSING, and the IEEE TRANSACTIONS ON COMPUTATIONAL BIOLOGY AND BIOINFORMATICS.

CHEN LI (Member, IEEE) received the Ph.D. degree from the University of Cambridge, in 2014. He is currently a Professor with the Department of Computer Science and Technology, Xi’an Jiaotong University. He is also holding a postdoctoral position at the Massachusetts Institute of Technology, in 2015. He is also working with the Department of Computer Science and Technology, School of Telecommunications, Xi’an Jiaotong University, where he is also a Professor and a Ph.D. Supervisor. His research interests include biomedical semantic understanding and natural language processing. He is the winner of Overseas Fellowship of Cambridge University, the Fellowship of European Molecular Biology Laboratory, the Thousand Talents Program of Central Organization Department, and the Young Top Talents Program of Xi’an Jiaotong University.

***