Stacked Residual Recurrent Neural Network with Word Weight for Text Classification

Wei Cao, Anping Song, Jinglu Hu,

***Abstract*—Neural networks, and in particular recurrent neural networks (RNNs) have recently been shown to give a state-of-the-art performance on some text classification tasks. However, most existing methods assume that each word in a sentence contributes the same importance, it is different from the real world. For example, if we do sentiment analysis, the word "awesome" is much more important than any other words in the sentence "This movie is awesome". Motivated by this deficiency and in order to achieve a further performance, in this paper, a Stacked Residual RNN with Word Weight method is proposed, we extend the stacked RNN to a deep one with residual network architecture and introduce a word weight based network to consider the weight of each word. Our proposed method is able to learn high the hierarchical meaning of each word in a sentence and consider the weight of each word for text classification task. Experimental result indicates that our method achieves high performance compared with the state-of-the-art approaches.**

***Index Terms*—Recurrent Neural Networks, Word Weight, Text Classification, Residual Networks, Long Short-Term Memory**

I. Introduction

TEXT classification is one of the main research areas in natural language processing, in which one needs to assign a label to each sentence. It has been widely used in some applications including sentiment analysis [1] [2], question type classification [3] [4] and topic type labeling [5] [6]. Sentence modeling is an important step in natural language processing for representing phrases and sentences into meaningful feature vectors which can be used for the classification task. Machine learning related methods have been widely used in text classification area.

The traditional methods for sentence modeling are based on bag-of-words (BOW) models, these methods concentrate on constructing hand-crafted features [1] such as representing words by one-hot vectors, and then use a linear model or kernel methods to classify text data [7]. But BOW models do not consider the order of words. For example, the sentence "Michael loves Jane" is different from the sentence "Jane loves Michael" while they have the same representations by using BOW model. Distributed representations were proposed by [8] and has become popular for been developed in the context of statistical language modeling by [9] which tried to represent each word in a dense and low dimensional vector. Moreover, [10] introduced an unsupervised method to learn representations from variable-length pieces of texts. Recently, neural networks, and in particular Recurrent Neural Networks (RNNs) [11] have been found successfully in learning text representations and been shown to give a state-of-the-art performance on some text classification tasks. RNNs are effective tools for sequence modeling tasks which are able to process a sequence of arbitrary length of sentence. RNNs with Long Short-Term Memory networks (LSTMs) structure [12] have a modified hidden state update which can more effectively capture long-term dependencies than standard RNNs. LSTMs have been widely used in many sequence modeling and prediction tasks, especially speech recognition [13], handwriting recognition [14] and machine translation [15].

Although recurrent neural networks based classifiers can get high performance in many text classification tasks [16] [17], one shortcoming of these neural text classification models is that they do not consider the different importance of each word in a sentence. For example, in the sentence "Today is so great", the word "great" is much more important than any other words for deciding polarity in sentiment analysis task.

In this paper, we propose a power of deep neural network approach for text classification. The method has a special architecture called Word Weight Network which can consider the contribution of different words in a sentence. Word Weight Network has the ability to learn the weight of word during the training procedure. Specifically, the output of the previous layer will not input to next layer directly in stacked RNN, the input of the previous layer will be trained in another network as word weight, and then it will multiply back to be the input of next layer. Besides, inspired by the high performance of Residual Networks (ResNets) [18] for training deep neural networks, we introduce a residual mechanism to improve the performance of stacked RNN. The Residual Network is an intriguing network which can overcome the disadvantage of vanishing gradients, exploding gradients and difficulties during the training process due to the increasing network depth in image recognition task.

The entire model is trained end-to-end with cross-entropy loss. The experiment results show that our model can achieve competitive accuracy. The main contributions of this work are as follows:

- We present a neural network approach which is able to learn the high hierarchical meaning of each word. It can learn different word weight from sentence during the training in text classification task.
- We demonstrate results on several text classification tasks. The empirical results show that our model can improve the accuracy of classification and outperforms state-of-the-art methods on three tasks.
- Our model can be trained end-to-end from input-output

Wei Cao is with the School of Computer Engineering and Science, Shanghai University, Shanghai, 200444 China and is also with the Graduate School of Information, Production and Systems, Waseda University, Fukuoka, 808-0135 Japan (e-mail: caowei@i.shu.edu.cn / caowei@toki.waseda.jp).

Anping Song is associate professor with the School of Computer Engineering and Science, Shanghai University, Shanghai, 200444 China (corresponding author, e-mail: apsong@shu.edu.cn).

Jinglu Hu is professor with the Graduate School of Information, Production and Systems, Waseda University, Fukuoka, 808-0135 Japan (e-mail: jinglu@waseda.jp).

pairs which mean that there is no additional artificial intervention required.

II. Related Works

A. Neural Networks

In recent years, deep learning has become famous in a wide variety of domains. It tries to simulate the human brain with artificial neural networks (ANNs) which can create a hierarchy of representations from data with its complex structures and multiple-layer models [19]. Convolutional Neural Networks (CNNs) [20] are a category of Neural Networks that have been proven to be able to achieve the best performance in computer vision [21] [22]. The convolution operation of CNNs can automatically capture the local dependencies from temporal or spatial data [23]. Recent research shows that CNNs can also be applied to NLP field, it can extract n-gram features at different positions of a sentence through convolutional filters and can learn short and long-range relations through pooling operations [24]. Recurrent Neural Networks (RNNs) [11] are a kind of neural networks which have the ability to map vectors of a sequence of arbitrary length to a fixed-length vector. RNNs use hidden state to save the memory of all the previous information from the sequence. RNNs are well suited to processing sequential data.

B. Text Classification

Classification task is the basic research in NLP area. A supervised classification algorithm allows us to access the data labels during the training and testing steps.

Deep learning based methods have achieved great results on text classification tasks. [10] proposed the Paragraph Vector method for representations of sentences and documents. [25] constructed a Character-level Convolutional Networks for doing text classification. [26] proposed a Convolutional Neural Networks for Sentence Classification which consider each word as n-gram to do embedding operation. [27] proposed a scheme for embedding learning of small text regions which is based on the idea of two-view semi-supervised learning. RNN models can achieve high performance on text related tasks. [28] first used RNNs for sequence text task. [29] propose Bidirectional Long Short-Term Memory with word embedding for text which contains richer syntactic and has a strong intrinsic dependency between words and phrases. [30] introduced a model to learn vector-based document representation in a unified, bottom-up fashion for sentiment classification. [31] utilized a Recurrent Convolutional Neural Networks method which use Convolutional and Recurrent Networks to capture the feature of contextual information to learn word representations. [32] proposed an intuitive approach to learn distributed word representations with Bi-LSTM.

We also mentioned that there are some novel methods for the related classification task. [33] introduced a method to expand short texts based on word embedding clustering and convolutional neural network. [34] used Multi-Task Learning methods to construct the model.

III. A Stacked Residual RNN Model

Our model assumes that each word in a sentence doesn't have the same importance, which means the output of the previous LSTM will not be the input of the next LSTM directly in stacked network.

In this model, it can consider the output weight of each word during the training, and inspired by the architecture of ResNets, we combine the idea of ResNets into our model for the sake of gradient vanish problem when the network is very deep.

The overall architecture of the Stacked Residual Recurrent Neural Network with Word Weight model (SRWW-RNN) is shown in Figure 1. As we can see, the left part of the model is called Word Weight Network part. This part takes responsibility for training the weight of each word. Utilizing the idea of Residual Networks, in the right part of model, we can see the input of the previous layer can add with the input of next layer directly, this is called short connections (The input and output are of the same dimensions). Therefore, the right part of this model is called Word Residual Network part.

A. Word Weight Network

Word Weight Network has the ability to learn the weight of each word. We believe that the word weight is very important in text categorization task, the label of the sentence is often determined by several key words. We focus on constructing this network by using fully-connected highway network [35] and Bidirectional LSTM (Bi-LSTM) [36]. The standard LSTM is updated as follows:

$$i_t = \sigma(W_i x_t + U_i h_{t-1} + b_i) \quad (1)$$

$$f_t = \sigma(W_f x_t + U_f h_{t-1} + b_f) \quad (2)$$

$$\tilde{c}_t = tanh(W_c x_t + U_c h_{t-1} + b_c) \quad (3)$$

$$c_t = f_t \odot c_{t-1} + i_t \odot \tilde{c}_t \quad (4)$$

$$o_t = \sigma(W_o x_t + U_o h_{t-1} + b_o) \quad (5)$$

$$h_t(LSTM) = o_t \odot tanh(c_t) \quad (6)$$

where x_t are the input of each time step t, W_j, U_j are the weight matrices and b_j are the bias vectors, for $j \in \{i, f, c, o\}$. σ denotes the sigmoid activation function and $\odot$ denotes element-wise multiplication. The forget gate controls how much of the previous state is going to be thrown away, the input gate controls how much of newly state will be updated, and the output gate controls how much of the internal memory state will be output.

The Bi-LSTM is a variant of LSTM. It contains not only the forward $\overrightarrow{LSTM}$ which reads the word from the beginning of a sentence to the end of a sentence but also the backward $\overleftarrow{LSTM}$ which reads the word from the end of a sentence to the beginning of a sentence:

$$\overrightarrow{h_t} = \overrightarrow{h_t(LSTM)} \quad (7)$$

$$\overleftarrow{h_t} = \overleftarrow{h_t(LSTM)} \quad (8)$$

$$h_{t,Bi-LSTM_W} = [\overrightarrow{h_t}, \overleftarrow{h_t}] \quad (9)$$

where $h_{t,Bi-LSTM_W}$ is the hidden state of the Bi-LSTM in word weight module which combines the forward and backward hidden states at each time step. Conventional standard LSTMs only utilize the previous context with no

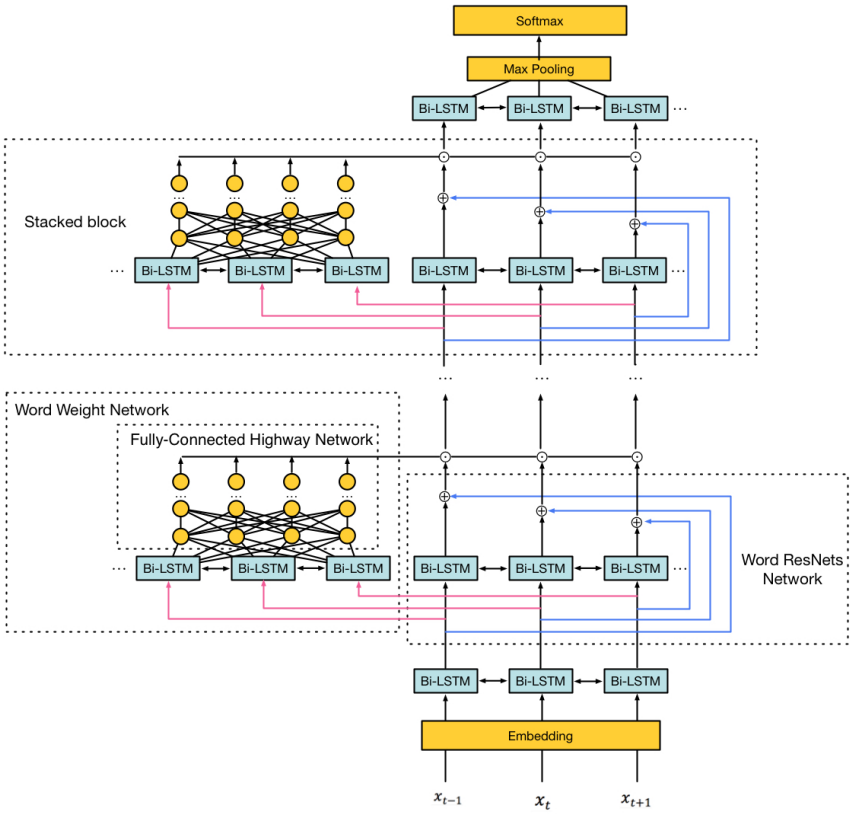

Fig. 1. The instance of Stacked Residual Bi-LSTM with Word Weight Networks. (The left part is the word weight training module and the right part is the word residual module.)

exploitation of future context, Bi-LSTMs utilize both the previous and future context.

The output of Bi-LSTM will be trained in fully-connected highway network as the word weight:

$$G_t = \sigma(Wh_{t,Bi-LSTM_W} + b) \quad (10)$$
$$H_t = ReLU(\tilde{W}h_{t,Bi-LSTM_W} + \tilde{b}) \quad (11)$$
$$C_t = 1 - G_t \quad (12)$$
$$O_t = H_t G_t + h_{t,Bi-LSTM_W} C_t \quad (13)$$

where W and $\tilde{W}$ are the weight matrices, b and $\tilde{b}$ are the bias vectors. $ReLU$[37] denotes the activation function. G_t is the transform gate, it can be used to control how much transformation of output is applied. C_t is the carry gate, this gate controls how much of the output can just be carried. O_t is the output of Word Weight Network.

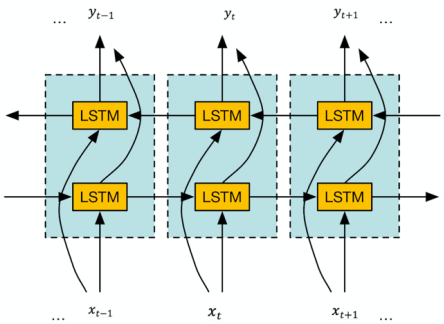

Fig. 2. An illustration of Bidirectional LSTM network.

B. Word Residual Network

The right part of Figure 1 shows the Word Residual Network of this model. As we can see, at each layer, the input of Bi-LSTM and the output of Bi-LSTM can be summed directly. The notion of Residual Networks (ResNets) was first introduced by [18] in image recognition area. The main idea of ResNets is to connect some of the layers with shortcuts, which can avoid vanishing gradients and exploding gradients problems, these problems may happen in very deep networks. With the increasing depth of networks, ResNets can improve the accuracy of deep networks.

The shortcut connections have the ability to explicitly let these layers fit a residual mapping with the help of identity transformation. The residual block defined as:

$$F(x_{i-1,t}) = O_t \quad (14)$$
$$x_{i,t} = ReLU(F(x_{i-1,t}) + id(x_{i-1,t})) \quad (15)$$

where $F(\cdot)$ function represents the Bi-LSTM transformation from $x_{i-1,t}$ layer to $x_{i,t}$ at each time step t, $id(\cdot)$ is an identity mapping function. $ReLU$ is the activation function for output of Word Residual block.

Although the derivation of Residual Networks is from image recognition area, inspired by its special architecture, we introduce it in our Stacked Bidirectional LSTM when the layers of Stacked Bi-LSTM are deep. The gradients and features which were learned in lower layers can pass through by the identity transformations $id(\cdot)$ in Word Residual Networks.

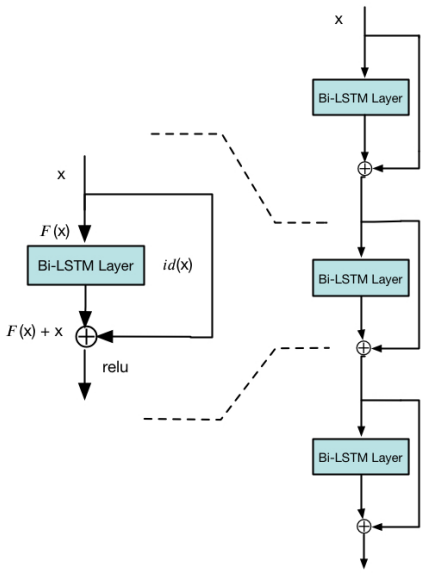

Fig. 3. The left image is a residual block in Word Residual Networks. The right image is an illustration of Word Residual Networks.

C. Stacked Residual Bi-LSTM with Word Weight Network

We extend our model to stacked one. Stacked based Bi-LSTM is the vertical multi-layer structure, the output of the lower layer will be the input of the upper layer. By using the stacked based structure, it is possible to achieve different levels of deep abstraction. There are some researches show that the deep hierarchical LSTM based model can be more efficient in representing some functions than a shallow one [38] [39].

The max-pooling vector of the output of Bi-LSTM can be used as the representation of the sentence. We add a linear transformation layer to transform vector to another vector which dimension is label number C. Then, we add $softmax$ layer to achieve conditional probabilities:

$$P_i = \frac{exp(x_i)}{\sum_{i'=1}^{C} exp(x_{i'})} \tag{16}$$

The target of the model is to predict label $\hat{y}_j^{(i)}$ for each sentence. We train the model over the training examples by minimizing the cross-entropy:

$$L(w) = \sum_{i=1}^{m} \sum_{k=1}^{K} 1\{y^{(i)} = k\} log(\hat{y}_j^{(i)}) \tag{17}$$

where $1\{\cdot\}$ is indicator function so that $1\{\text{true}\}$=1, and $1\{\text{false}\}$=0. m is the number of training examples. $y^{(i)} \in \{1, 2, \ldots K\}$ is true label of each sentence and K is the number of possible labels. $\hat{y}_j^{(i)} \in [0, 1]$ is estimated probabilities of each sentence of each label.

We use Adam[40] stochastic gradient descent optimizer to update the parameters and use accuracy metric to evaluate the performance of our model:

$$accuracy = \frac{TN + TP}{TN + TP + FN + FP} \tag{18}$$

where:

TP: correctly predicted positive samples.

TN: correctly predicted negative samples.

FP: positive samples that incorrectly predicted.

FN: negative samples that incorrectly predicted.

IV. Experimental Setup

A. Datasets

To show the effectiveness of our model, we choose four different text classification tasks to evaluate our Stacked Residual with Word Weight architecture.

SST-1: The movie reviews consist of 11855 movie reviews with five labels: very negative, negative, neutral, positive, and very positive in Stanford Sentiment Treebank[41]. The dataset is split into train (8544), dev (1101), and test (2210) for the fine-grained classification task.

SST-2: The movie reviews with binary labels by removing neural labels from the Stanford Sentiment Treebank. The dataset is split into train (6920), dev (872), and test (1821) for the binary classification task.

TREC: We choose the TREC [4] which is a question type classification benchmark. TREC consists of 6 classes, including location, human, entity, abbreviation, description and numeric. The training dataset contains train (5452) and test (500) questions.

SUBJ: Subjectivity dataset where the goal of task is to classify each sentence as being subjective or objective [42].

B. BaseLines

We compare our model with several models as follow:

SVM: SVM with unigram and bigram features [41] [43].

NBOW: NBoW averages word vectors and applies a softmax classification layer [44].

Paragraph Vector: Paragraph Vector learns fixed-length feature representations from variable-length pieces of texts [10] [45].

CNN-non-static: Convolutional Neural Network based model with fine-tuned word vectors [26].

CNN-multichannel: Convolutional Neural Network based model with multi-channels [26].

DCNN: Dynamic Convolutional Neural Network with dynamic k-max pooling [44].

RAE: Recursive autoencoder learns vector space representations for multi-word phrases [41].

MV-RNN: Matrix-Vector Recursive Neural Network with parse trees which can represent every word and longer phrase in a parse tree as both a vector and a matrix [41].

RNTN: Recursive Neural Tensor Network is able to use the same, tensor-based composition function for all nodes [41].

DRNN: Multi-layer stacked Recursive Neural Network [46].

Multi-Task: A multi-task learning framework to jointly learn across multiple related tasks. This model can share information among these tasks. [34].

Tree-LSTM: A generalization of LSTMs to tree structured network topologies [47].

C-LSTM: C-LSTM extract a sequence of higher-level phrase representations from CNN and fed them into LSTM to obtain the sentence representations [24].

Gaussian: A method which models each document as a Gaussian distribution based on the embeddings of its words [48].

COMB: A linguistically-motivated approach that corrupts training examples with linguistic noise [49].

TABLE I
THE STATISTICAL DETAIL OF FOUR DATASETS IN OUR EVALUATION

Dataset	Class	Train Size	Valid Size	Test Size	Average Length	Max Length	Vocabulary Size
SST-1	5	8544	1101	2210	19.1	56	19.5k
SST-2	2	6920	872	1821	19.3	56	17.5k
TREC	6	5452	-	500	9.9	37	8.9k
SUBJ	2	9000	-	1000	22.0	113	21k

TABLE II
SOME OTHER HYPERPARAMETERS SETTINGS AMONG FOUR DATASETS

Hyperparameters	SST-1 / SST-2	TREC	SUBJ
Memory dimension (Bi-LSTM)	150	300	200
Stacked blocks	8	3	4
Hidden layers (Fully-connected highway network)	5	5	5
Hidden units (Fully-connected highway network)	50	50	50

C. Hyperparameters and Training

In our experiments, we initialize word embeddings with the publicly available 300-dimensional word vectors. The vectors are pre-trained with word2vec on Google News Dataset which contains about 100B words [50] [51]. We also initialize the vector with the uniform distribution [-0.25, 0.25] for words which are not in word2vec vectors.

We train our model with Adam stochastic gradient descent optimizer with a learning rate of 0.001 and we use a mini-batch size of 50. The parameters are initialized from the uniform distribution in [-0.1, 0.1]. The parameters were regularized with L2 regularization with the factor of 10^{-4}. We also apply dropout[52] with a probability of 0.5 on both Word Weight Network and Word Residual Network during the training to prevent overfitting.

Other hyperparameters settings are shown in Table II, for SST, we set the hidden layer size of LSTM is 150, so the combination of forward and backward network gives us 300-dimension vectors in Bi-LSTM. The same as TREC.

D. Results and Analysis

The experimental results are showed in Table III. We compare our model with a variety of models, the Stacked Residual with Word Weight structure model has high performance on text classification task without any additional feature engineering.

For SST and SUBJ dataset, our proposed method outperforms existing models and achieves state-of-art prediction accuracy. In particular, our model obtains 52.7% classification accuracy on fine-grained classification task which is a very substantial improvement. For TREC, our result is close to the best accuracy. Although we did not beat the state-of-the-art one, comparing with SST and SUBJ, we find that not only the average sentences length of SST and SUBJ are longer than TREC, but also the semantic complexity of them are much more complicated than TREC. Through the analysis,

Algorithm 1 The pseudo-code of our model

Input: Sentences
Output: Label of sentences

Pre-train sentences with word2vec to generate word vectors.
for $n = 1 \rightarrow N$ (N represents the number of sentences) **do**
    for $m = 1 \rightarrow M$ (M represents the number of words in current sentence) **do**
        Do word embedding to obtain the vector of each word.
        Employ Bi-LSTM to obtain the sequence representations.
        for $k = 1 \rightarrow K$ (K represents the number of layers of Stacked Bi-LSTM) **do**
            The output of the previous layer would input to next Bi-LSTM to obtain the hierarchical sequence representations.
            The output of the previous layer would input to Word Weight Network to obtain the weight information (Word Weight Network).
            The output of step 7 would calculate the element-wise product with the output of step 8.
            The output of step 9 would add with the input of step 7 as new sequence representations (Word Residual Network).
        end for
    end for
    Employ Softmax classifier to get the label of each sentence.
    Update parameters by using the stochastic gradient descent algorithm.
end for

TABLE III
CLASSIFICATION ACCURACY (%) OF OUR METHOD COMPARED WITH OTHER MODELS ON FOUR DATASETS

Methods	SST-1	SST-2	TREC	SUBJ
SVM	40.7%	79.4%	95%	-
NBOW	42.4%	80.5%	-	-
Paragraph Vector	48.7%	87.8%	91.8%	-
CNN-non-static	48.0%	87.2%	93.6%	93.4%
CNN-multichannel	47.4%	88.1%	92.2%	93.2%
DCNN	48.5%	86.8%	93.0%	-
RAE	43.2%	82.4%	-	-
MV-RNN	44.4%	82.9%	-	-
RNTN	45.7%	85.4%	-	-
DRNN	49.8%	86.6%	-	-
Multi-Task	49.6%	87.9%	-	94.1%
Tree-LSTM	51.0%	88.0%	-	-
C-LSTM	49.2%	87.8%	94.6%	-
Gaussian	-	-	**98.2%**	93.1%
COMB	-	84.8%	-	93.6%
SRWW-RNN	**52.7%**	**88.2%**	95.6%	**95.0%**

we think that our model is more applicable to the sentence which has complex semantics. The Stacked Residual with Word Weight structure has the ability to learn the weight of different words, it is very useful for sentence representation that can increase the prediction accuracy. The results mean that our model can extract more information and learn high hierarchical features from the text than other approaches on the dataset which has complex semantics.

As we can see from Figure 4(a), comparing with the standard stacked Bi-LSTM. During the training step, the convergence speed of our model is faster. The inputs of a lower layer in stacked Bi-LSTM are made available to a node in a higher layer because of the shortcut connections which can lead the network easy trained. The Figure 4(b) shows the test accuracy between two models. this figure indicates that the Word Residual Network is able to achieve high accuracy and the gradients can easily back propagate through them, which results in a faster converging.

In order to show the comprehensive performance, we do another experiment on SUBJ dataset to compare our model with the standard recurrent models and the bidirectional recurrent models. Table IV shows the comparison results on precision, recall and f1-score metrics. According to the results, we find that LSTM based models have better performance than RNN based models. The gate mechanism can control the flow of information and cell state. Besides, the bidirectional structures which combine both forward and backward layers have higher accuracy than standard structures. We also compare our model initialized with random word embedding and pre-trained word2vec. Figure 5 illustrates the comparison results which indicate that pre-trained word embedding method can learn meaningful information from context well.

Benefiting from the word weight structure, our model has

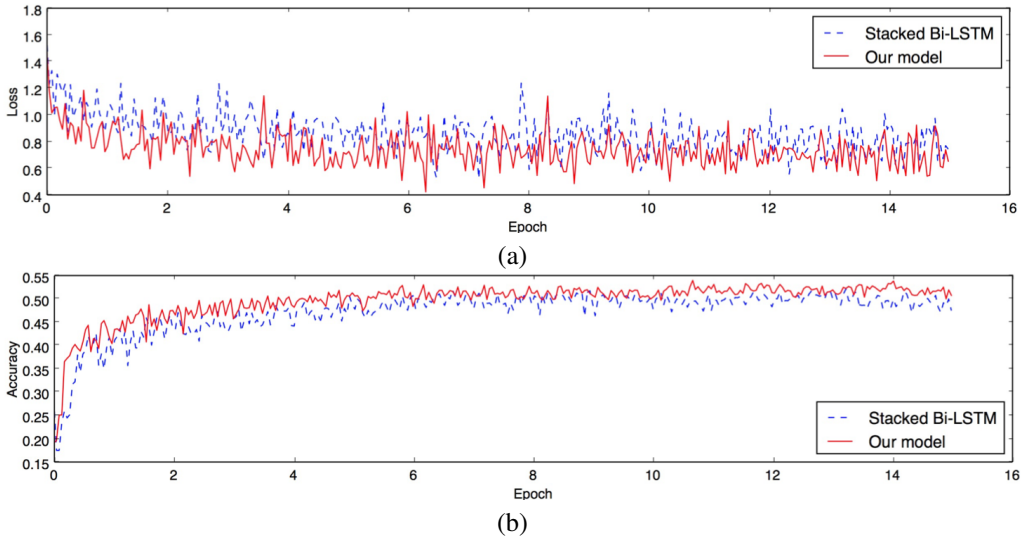

Fig. 4. The (a) shows the batch training loss on SST-1 with our method and standard stacked Bi-LSTM. The (b) shows the test accuracy on SST-1 compared with two methods.

TABLE IV
COMPREHENSIVE EVALUATION COMPARISON RESULTS (%) ON SUBJ DATASET

Methods	Precision	Recall	F1-score
RNN	86.25%	91.52%	88.80%
Bi-RNN	89.94%	93.49%	91.68%
LSTM	94.29%	94.47%	94.38%
Bi-LSTM	94.83%	94.08%	94.46%
SRWW-RNN	**94.89%**	**95.27%**	**95.08%**

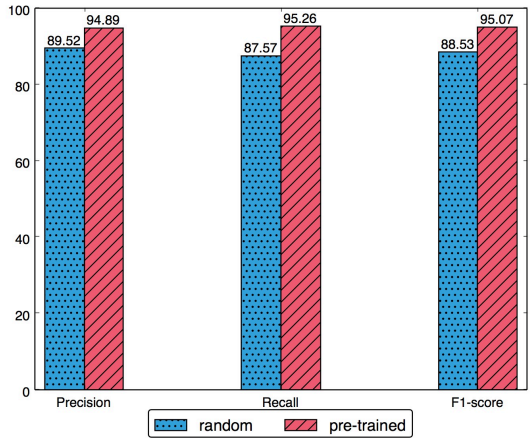

Fig. 5. Precision, Recall and F1-score (%) comparison by impact of initialized word embedding on SUBJ dataset.

the ability to learn the importance of different words. We select two sentences from the SST-2 dataset and visualize the probability change of output layer at different time steps. In Figure 6, for the sentence "As a singular character study, it 's perfect.", which has a positive label. We find that our model has high sensitivity on keywords such as "singular" and "perfect". Most of the time, the label of each sentence is determined by these keywords.

V. CONCLUSIONS

In this paper, we introduce a novel text classification model called Stacked Residual Recurrent Neural Network with Word Weight. The Word Weight Network can identify the contribution of different words due to its special structure. The Word Residual Network makes the model more expressive when stacked layers are deep. The new architecture has the ability to extract more features and learn the high hierarchical meaning of each word from a sentence. This model can be applied to some natural language processing tasks such as analyzing the implicit semantic information of words. We tried to demonstrate the effectiveness of our model by applying it to text classification task. Experimental results show that our model can achieve better performance than previous methods. This suggests our model can capture more potential features in sentences.

ACKNOWLEDGMENT

This research is supported by the Major Research plan of the National Natural Science Foundation of China (Grant No. 91630206)

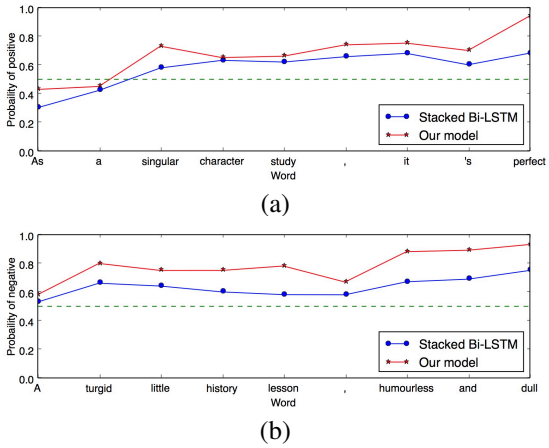

Fig. 6. The (a) shows the probability change of positive label at different time steps. The (b) shows the probability change of negative label at different time steps.

REFERENCES

[1] A. L. Maas, R. E. Daly, P. T. Pham, D. Huang, A. Y. Ng, and C. Potts, "Learning word vectors for sentiment analysis," in *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1*. Association for Computational Linguistics, 2011, pp. 142–150.

[2] M. Ghiassi, J. Skinner, and D. Zimbra, "Twitter brand sentiment analysis: A hybrid system using n-gram analysis and dynamic artificial neural network," *Expert Systems with applications*, vol. 40, no. 16, pp. 6266–6282, 2013.

[3] D. Zhang and W. S. Lee, "Question classification using support vector machines," in *Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval*. ACM, 2003, pp. 26–32.

[4] X. Li and D. Roth, "Learning question classifiers," in *Proceedings of the 19th international conference on Computational linguistics-Volume 1*. Association for Computational Linguistics, 2002, pp. 1–7.

[5] S. Wang and C. D. Manning, "Baselines and bigrams: Simple, good sentiment and topic classification," in *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Short Papers-Volume 2*. Association for Computational Linguistics, 2012, pp. 90–94.

[6] J.-M. Yang, Z.-Y. Liu, and Z.-Y. Qu, "Clustering of words based on relative contribution for text categorization," *IAENG International Journal of Computer Science*, vol. 40, no. 3, pp. 207–219, 2013.

[7] T. Mullen and N. Collier, "Sentiment analysis using support vector machines with diverse information sources." in *EMNLP*, vol. 4, 2004, pp. 412–418.

[8] G. E. Hinton, "Learning distributed representations of concepts," in *Proceedings of the eighth annual conference of the cognitive science society*, vol. 1. Amherst, MA, 1986, p. 12.

[9] Y. Bengio, R. Ducharme, P. Vincent, and C. Jauvin, "A neural probabilistic language model," *Journal of machine learning research*, vol. 3, no. Feb, pp. 1137–1155, 2003.

[10] Q. V. Le and T. Mikolov, "Distributed representations of sentences and documents." in *ICML*, vol. 14, 2014, pp. 1188–1196.

[11] J. L. Elman, "Finding structure in time," *Cognitive science*, vol. 14, no. 2, pp. 179–211, 1990.

[12] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[13] X. Li and X. Wu, "Constructing long short-term memory based deep recurrent neural networks for large vocabulary speech recognition," in *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2015, pp. 4520–4524.

[14] T. M. Breuel, A. Ul-Hasan, M. A. Al-Azawi, and F. Shafait, "High-performance ocr for printed english and fraktur using lstm networks," in *2013 12th International Conference on Document Analysis and Recognition*. IEEE, 2013, pp. 683–687.

[15] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," in *Advances in neural information processing systems*, 2014, pp. 3104–3112.

[16] J. Y. Lee and F. Dernoncourt, "Sequential short-text classification with recurrent and convolutional neural networks," in *Proceedings of NAACL-HLT*, 2016, pp. 515–520.

[17] J. Zhou and W. Xu, "End-to-end learning of semantic role labeling using recurrent neural networks," in *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, 2015.

[18] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 770–778.

[19] R. Bustami, N. Bessaih, C. Bong, and S. Suhaili, "Artificial neural network for precipitation and water level predictions of bedup river," *IAENG International Journal of computer science*, vol. 34, no. 2, pp. 228–233, 2007.

[20] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[21] K. Tang, M. Paluri, L. Fei-Fei, R. Fergus, and L. Bourdev, "Improving image classification with location context," in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 1008–1016.

[22] S. Zhang, J. Wang, X. Tao, Y. Gong, and N. Zheng, "Constructing deep sparse coding network for image classification," *Pattern Recognition*, vol. 64, pp. 130–140, 2017.

[23] T. L. Li, A. B. Chan, and A. H. Chun, "Automatic musical pattern feature extraction using convolutional neural network," in *Proceedings of the International MultiConference of Engineers and Computer Scientists*, vol. 1, 2010.

[24] C. Zhou, C. Sun, Z. Liu, and F. Lau, "A c-lstm neural network for text classification," *arXiv preprint arXiv:1511.08630*, 2015.

[25] X. Zhang, J. Zhao, and Y. LeCun, "Character-level convolutional networks for text classification," in *Advances in Neural Information Processing Systems*, 2015, pp. 649–657.

[26] Y. Kim, "Convolutional neural networks for sentence classification," *arXiv preprint arXiv:1408.5882*, 2014.

[27] R. Johnson and T. Zhang, "Semi-supervised convolutional neural networks for text categorization via region embedding," in *Advances in neural information processing systems*, 2015, pp. 919–927.

[28] T. Mikolov, M. Karafiát, L. Burget, J. Cernockỳ, and S. Khudanpur, "Recurrent neural network based language model." in *Interspeech*, vol. 2, 2010, p. 3.

[29] Z. Xiao and P. Liang, "Chinese sentiment analysis using bidirectional lstm with word embedding," in *International Conference on Cloud Computing and Security*. Springer, 2016, pp. 601–610.

[30] D. Tang, B. Qin, and T. Liu, "Document modeling with gated recurrent neural network for sentiment classification," in *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, 2015, pp. 1422–1432.

[31] S. Lai, L. Xu, K. Liu, and J. Zhao, "Recurrent convolutional neural networks for text classification." in *AAAI*, 2015, pp. 2267–2273.

[32] P. Wang, Y. Qian, F. K. Soong, L. He, and H. Zhao, "Learning distributed word representations for bidirectional lstm recurrent neural network," in *Proc. of ICASSP*, 2016.

[33] P. Wang, B. Xu, J. Xu, G. Tian, C.-L. Liu, and H. Hao, "Semantic expansion using word embedding clustering and convolutional neural network for improving short text classification," *Neurocomputing*, vol. 174, pp. 806–814, 2016.

[34] P. Liu, X. Qiu, and X. Huang, "Recurrent neural network for text classification with multi-task learning," *arXiv preprint arXiv:1605.05101*, 2016.

[35] R. K. Srivastava, K. Greff, and J. Schmidhuber, "Training very deep networks," in *Advances in neural information processing systems*, 2015, pp. 2377–2385.

[36] A. Graves and J. Schmidhuber, "Framewise phoneme classification with bidirectional lstm and other neural network architectures," *Neural Networks*, vol. 18, no. 5, pp. 602–610, 2005.

[37] V. Nair and G. E. Hinton, "Rectified linear units improve restricted boltzmann machines," in *Proceedings of the 27th International Conference on Machine Learning (ICML-10)*, 2010, pp. 807–814.

[38] A. Graves, A.-r. Mohamed, and G. Hinton, "Speech recognition with deep recurrent neural networks," in *2013 IEEE international conference on acoustics, speech and signal processing*. IEEE, 2013, pp. 6645–6649.

[39] J. Chung, C. Gülçehre, K. Cho, and Y. Bengio, "Gated feedback recurrent neural networks," *CoRR, abs/1502.02367*, 2015.

[40] D. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[41] R. Socher, A. Perelygin, J. Y. Wu, J. Chuang, C. D. Manning, A. Y. Ng, and C. Potts, "Recursive deep models for semantic compositionality over a sentiment treebank," in *Proceedings of the conference on empirical methods in natural language processing (EMNLP)*, vol. 1631. Citeseer, 2013, p. 1642.

[42] B. Pang and L. Lee, "A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts," in *Proceedings of the 42nd annual meeting on Association for Computational Linguistics*. Association for Computational Linguistics, 2004, p. 271.

[43] J. Silva, L. Coheur, A. C. Mendes, and A. Wichert, "From symbolic to sub-symbolic information in question classification," *Artificial Intelligence Review*, vol. 35, no. 2, pp. 137–154, 2011.

[44] N. Kalchbrenner, E. Grefenstette, and P. Blunsom, "A convolutional neural network for modelling sentences," *arXiv preprint arXiv:1404.2188*, 2014.

[45] H. Zhao, Z. Lu, and P. Poupart, "Self-adaptive hierarchical sentence model," *arXiv preprint arXiv:1504.05070*, 2015.

[46] O. Irsoy and C. Cardie, "Deep recursive neural networks for compositionality in language," in *Advances in Neural Information Processing Systems*, 2014, pp. 2096–2104.

[47] K. S. Tai, R. Socher, and C. D. Manning, "Improved semantic representations from tree-structured long short-term memory networks," *arXiv preprint arXiv:1503.00075*, 2015.

[48] G. Nikolentzos, P. Meladianos, F. Rousseau, M. Vazirgiannis, and Y. Stavrakas, "Multivariate gaussian document representation from word embeddings for text categorization," *EACL 2017*, p. 450, 2017.

[49] Y. Li, T. Cohn, and T. Baldwin, "Robust training under linguistic adversity," *EACL 2017*, p. 21, 2017.

[50] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," *arXiv preprint arXiv:1301.3781*, 2013.

[51] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," in *Advances in neural information processing systems*, 2013, pp. 3111–3119.

[52] G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, and R. R. Salakhutdinov, "Improving neural networks by preventing co-adaptation of feature detectors," *arXiv preprint arXiv:1207.0580*, 2012.