Journal of Computers Vol. 33 No. 1, February 2022, pp. 165-178
doi: 10.53106/199115992022023301015

Multiple Scene Sentiment Analysis Based on Chinese Speech and Text

Haiyuan Guo*, Xuegang Zhan, Chengying Chi

University of Science and Technology Liaoning, Anshan, 114000, China
18841270927@163.com
Received 6 June 2021; Revised 17 August 2021; Accepted 8 September 2021

Abstract. This paper proposes a multi-scene sentiment analysis model for Chinese speech and text based on CNN-BiGRU-CTC + ERNIE-BiLSTM. The model is applied to the intelligent customer service scenario. While conducting voice interaction, intelligent customer service can obtain the user's current emotion, to give a more humane answer and improve the user experience. All the training data sets in this paper adopted public data sets such as Aishell-1 and NLPCC 2014, etc.We have been able to achieve a testing accuracy of about 94.5%. The accuracy is improved by 5.24% compared to the latest speech sentiment analysis model that uses audio as a feature. The advantage of this paper is that it adopts the ERNIE language pre-training model to conduct sentiment analysis on speech signals, which still has a good classification accuracy in the case of individual wrong words in speech recognition.

Keywords: ERNIE, bi-directional long-short term memory, convolutional neural networks, CTC, multi-scene, bidirectional gated recurrent unit, intelligent customer service

1 Introduction

Speech emotion recognition [1-2] (SER) refers to the emotional state of Speech signals that can be automatically recognized by a computer. As one of the main communication media of human beings, speech not only carries semantic information but also contains the speaker's emotional information. Making machines aware of human emotions can help to have a more natural and harmonious conversation in human-computer interaction. Enabling machines with the ability to recognize speech emotions can further improve the performance of speech recognition and speaker recognition, which is the key to realizing natural human-computer interaction [3-4].

Sentiment feature extraction, as an important part of speech emotion recognition, has attracted extensive attention from many researchers. Most of these studies focus on emotion recognition involving some of the most distinctive manual features. More specifically, feature extraction consists of two stages. First of all, acoustic features are extracted from each burst of speech signals, usually including prosody features, spectral-based related features, sound quality features, and non-linear features applied to each speech to graduate features and statistical features. Searching for traits that show a high degree of emotional correlation through several carefully prepared experiments can be a time-consuming and laborious task. In addition, the effectiveness of the selected features is still largely dependent on the pattern recognition model shown, resulting in low generality. Recently, a new trend has emerged in the field of deep learning, which directly utilizes deep neural networks to extract speech features, and sets to obtain emotional representations of input signals directly from raw, unprocessed speech data. The reason behind the idea is that the network can automatically learn the intermediate representation of the raw voice signal, making it better adapted to the task at hand, which in turn leads to improved performance. In this regard, end-to-end [5-6] learning has become a very progressive option. A large number of end-to-end learning frameworks have been rapidly and widely applied in speech emotion recognition. George Trigeorgis et. [7] proposed a context-aware emotion-related feature extraction method based on the combination of CNN(Convolutional Neural Network) and LSTM(Long-Short Term Memory RNN) network, to automatically obtain the best representation of speech signals from the original data. In the Recola database, the recognition rate of "Arousal" was 68. 60%, and the recognition rate of "Valence" was 26. 10%. Experimental results indicate that the topology is significantly better than the conventional topology. Siddique Latif et. [8] proposed an end-to-end training model, in which long-term and short-term interactions were directly captured from the original speech by the parallel convolution layer, and different context dependencies were captured by the feature graph output by LSTM to CNN. The recognition rate was 60. 23% in the IEMOCAP database. Pengcheng Li et. [9] learned emo-

* Corresponding Author

tional representation in an end-to-end way by directly applying CNN to the spectrogram extracted from speech. Two sets of filters of different shapes are designed to capture time and frequency domain information from the input sonograms. The weighted recognition rate on the IEMOCAP database was 71. 80%. Wootaek Lim et. [10] proposed a speech emotion recognition method based on tandem CNN and RNN(Recurrent Neural Networks), which combines the deeply layered CNNS feature extraction framework with the LSTM network layer. Instead of using any traditional manual features to classify emotional speech, the spectrogram is used as the input. The recognition rate on the EMO-DB database is 88. 01%, which verifies the feasibility of the end-to-end neural network model.

In this paper, a new end-to-end CNN-BiGRU-CTC + ERNIE-BiLSTM model is proposed based on traditional speech emotion recognition methods and driven by existing end-to-end deep neural network research progress. The network model takes into account the advantages of both CNN and BiGRU(Bidirectional gated recurrent unit) neural networks. The Convolutional Neural Network is used to learn local features in the time domain and frequency domain, and then a bidirectional gated loop unit network is added to learn context features. By using ERNIE's language pre-training model, the model based on the character input not only enhances the semantic representation of the word but also preserves the context information and the poly-meaning of the word. To make full use of the emotional information in the text, we learn the contextual information through the BiLSTM (Bi-directional long-short term memory) neural network.Our model achieves 94. 5% sentiment polarity classification accuracy, which is 5.24% better than the latest speech sentiment analysis models. The reason for such a good classification accuracy is that a new idea is used to solve the speech sentiment analysis problem by combining speech recognition and text sentiment analysis to achieve multi-scene sentiment analysis. Using ERNIE-BiLSTM model, it can still have good classification accuracy and good robustness in the case of individual spelling and pronunciation errors in the text acquired after speech recognition. Moreover, unlike the traditional speech emotion model, the model in this paper can obtain text data information, which can obtain semantic emotion information. Through experimental verification, good progress has been made so far, and the excellent accuracy of the model can be commercially used in the field of intelligent customer service to improve the accuracy, comfort and user experience of message replies.

2 Model Structure

The neural network model designed in this paper is composed of two parts: the speech recognition model and the emotion recognition model. The model structure is shown in Fig. 1. Firstly, the model preprocesses and extracts the features of the original Chinese audio sequence. Then through three CNN layers [11-14], each CNN layer involves four steps: convolution, batch normalization [15], rectifying linear unit [16]activation, and maximum pooling. The features extracted by CNN are used as an input of the BiGRU cell layer to enhance the speech sequence information representation, and then add the connection layer. The output at each time step output mapping for different characters of probability, finally by CTC(Connectionist temporal classification) [17] layer for decoding, output tags sequence y. The text y after speech recognition enters the emotion recognition model, and the ERNIE (Enhanced Language Representation with Informative Entities) training model first processes the text processed by word separation to obtain the corresponding word vectors, and then the obtained word vectors are used as the input of the BiLSTM model to further obtain the contextual information of each word. Finally, through the softmax layer vector for the classification of positive and negative emotions.

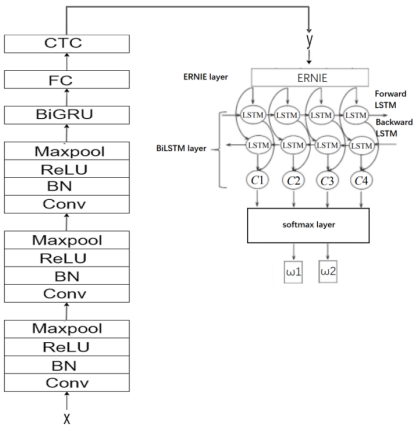

Fig. 1. Model structure diagram

2.1 Feature Extraction

In the end-to-end speech recognition system, the most commonly used features are Mel-frequency Cepstral Coefficient (MFCC) and Filter Bank (FBank). The MFCC is designed based on the auditory characteristics of human ears and is the cepstrum coefficient extracted at the frequency of the Meier scale. In this paper, MFCC is used for research and experiments, and the flow chart of feature extraction is shown in Fig. 2.

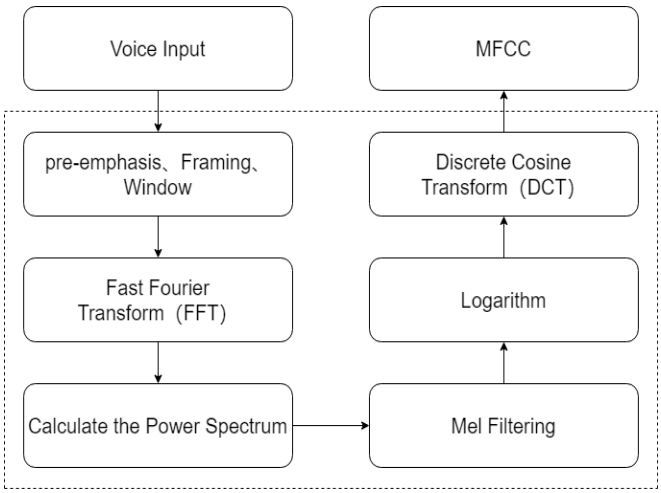

Fig. 2. Audio feature extraction process

2.2 Convolutional Neural Network Layer

Given an input sequence $X = \{ x_1, ..., x_T \}$, $x_i \in R^{b \times c}$, a convolution kernel $K \in R^{w \times h \times c}$, and a convolution step $SC = (sw_c, sh_c)$. The result of convolution is a two-dimensional feature graph, and its calculation method is shown in Equation (1):

$$o_{i,j} = \sum_{w_i, h_j, q} x_{sw_c} \cdot i + w_{i, sh_c} \cdot j + h_{j,q} \cdot k_{w_i, h_j, q} . \quad (1)$$

The time step, bandwidth and channel of the input sequence are T, b, c respectively, the breadth and depth of the kernel are w, h respectively, and the convolution kernel K is a matrix. The width step and height step of the convolution are sw_c, sh_c respectively. The Mel-frequency Cepstral Coefficient is the input sequence $\{ x_1, ..., x_T \}$.

According to Equation (1), each result element $o_{i,j}$ after convolution can be derived from the local elements of the input feature map $w \times h$, which means that the convolutional neural network layer can capture the local features of data. Therefore, for the convolution of the input sequence with c feature maps, each element is related to the local input elements of $c \times w \times h$.

Convolution learning local features is very useful in speech recognition tasks. In ASR, speech is viewed as a sequence of hundreds of subtle audio fragments, one after another. Learning local features of short audio is therefore an important step in speech recognition tasks. Single CNN can only cover a very small input range. If we stack multiple CNN layers together, it will be able to learn a larger range of local features.

In the case of just the timeline, we assume that the width of the CNN kernel is w_c and the width step is sw_c for the sake of explanation. We will refer to the window of time transfer as $window_c$ and the span of time covered

by the result element as t_c between two adjacent result elements. then $window_c$ and t_c are calculated from Equation (2):

$$t_c = t_i + (w_c - 1) \cdot window_i$$
$$window_c = sw_c \cdot window_i \quad \textbf{(2)}$$

Where a $window_i$ and t_i are the shift window and time range of the input.

For this paper, Meyer Frequencies Cepstrum Coefficients (MFCC) trains are used for input. Each of the MFCC frames has a 25ms time span with a 10ms moving window. In the time dimension, the kernel widths of the three layers of CNN are 3, 2 and 2 respectively. They both convolve with one step. Therefore, without the pool layer, the final result element CNN layer covers the time span of 65ms and transforms two adjacent elements between the window of 10ms, which means that the feature learned by layer 3 CNN is 65ms, far greater than the time span of the original MFCC frame.

2.3 Lstm and Gru

Both short-term and long-term memory network [18] (LSTM) and the door control cycle unit network (GRU helped) belong to a cycle of neural network. They are all in order to improve the iterative gradient diffusion or gradient explosion cycle neural network [19], and that the problem of LSTM long time memory unit Cell are introduced, and control of the Cell by gating mechanism information stored or not. LSTM model consists of three door control unit: forgotten door, input door and output. The left in figure LSTM figure 3 f, i, o, moment on the long-term memory of oblivion gate control unit information is forgotten;The inputs control whether the input information is input to the attempted memory unit information;Outputs control whether or not the information of the memory unit is output.

The GRU network is an improvement on the basis of the LSTM network. Since the inputs in the LSTM network and the forgetting gate are complementary, the two gates are merged into one gate in the GRU network-the update gate. In addition, GRU network merges the long-term memory unit Cell with the current state, directly establishing the linear dependency relationship between the current state and the historical state. With the same efficiency as LSTM network, the improved GRU network simplifies the network structure, reduces network parameters and has better convergence.

The LSTM is calculated using the following information:

x_t : Enter the data at time t.

h_{t-1} : The hidden state at time t-1.

c_{t-1} : Cell state at time t.

Given x_t, h_{t-1}, and c_{t-1}, the LSTM prioritizes the computation of the forgetting gate f_t (as Equation (3)), the input gate (as Equation (4)), the output gate (as Equation (5)), and the candidate context c_t (as Equation (6)).

$$f_t = \sigma([x_t; h_{t-1}]W_f + b_f) . \quad \textbf{(3)}$$
$$i_t = \sigma([x_t; h_{t-1}]W_i + b_i) . \quad \textbf{(4)}$$
$$o_t = \sigma([x_t; h_{t-1}]W_o + b_o) . \quad \textbf{(5)}$$
$$\tilde{c}_t = F_{\tilde{c}}([x_t; h_{t-1}]W_{\tilde{c}} + b_{\tilde{c}}) . \quad \textbf{(6)}$$

Then, LSTM calculates cell state c_t under the current step according to f_t, c_{t-1}, i_t and $\tilde{c}_t$, as shown in Equation (7).

$$c_t = f_t c_{t-1} + i_t \tilde{c}_t . \quad \textbf{(7)}$$

The LSTM then calculates the hidden state of the current step using o_t and c_t, It can be calculated according to Equation (8).

$$h_t = o_t * F(c_t) \ . \tag{8}$$

Finally, the hidden state h_t is the same as the output y_t at time t given by LSTM.

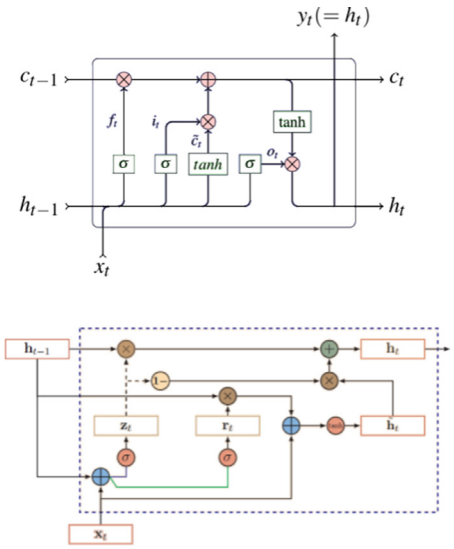

Fig. 3. LSTM (top) and GRU (bottom) structure diagram

The GRU calculates using the following information:

x_t : Enter the data at time t.

h_{t-1} : The hidden state at time t-1.

Given that x_t and h_{t-1} first calculate reset gate r_t, reset gate r_t is used to control whether the calculation of $\tilde{h}_t$ depends on state h_{t-1} at the previous moment.

$$r_t = \sigma\ (W_t x_t + U_t h_{t-1}) \ . \tag{9}$$

The candidate state at the current moment is:

$$\tilde{h}_t = \tanh\ ((W x_t + U\ (r_t \cdot h_{t-1}) \ . \tag{10}$$

The update gate z_t is then calculated, and the update gate z_t controls how much information the current state h_t needs to retain from the historical state h_{t-1} and how many new messages it needs to receive from the candidate state $\tilde{h}_t$.

$$z_t = \sigma(W_z x_t + U_z h_{t-1}) \ . \tag{11}$$

Then calculate the hidden state h_t :

$$h_t = (1 - z_t) \circ h_{t-1} + z_t \circ \tilde{h}_t \ . \tag{12}$$

There are many temporal dependencies in audio and text data. However, some of these data may have longer periods, and neither CNN nor the maximum pooling layer can capture them. Therefore, we better solve this problem by using GRU and LSTM layers.

However, in order to take full advantage of the historical and future information of the input features over the entire time range, we built a bidirectional GRU by superimposing two GRU layers in opposite directions, which maintained both forward and backward time.

2.4 Connectionist Temporal classification

Before CTC, there were some difficulties in end-to-end speech recognition. In the traditional acoustic model training of speech recognition, it is necessary to know the corresponding label for each frame data in order to carry out effective training. Before training the data, the speech needs to be pre-processed, which is a task that is very labor-intensive and time-consuming. This makes it difficult to build large databases. In addition, building a performing ASR is a difficult process because it requires a variety of expertise to design the modules.

The CTC algorithm can be broadly divided with two steps: probability calculation of paths and path merging. CTC introduces a blank tag to represent the mute frame and character interval, which we use in this article as ' - ', meaning that there is no output, only a path with an intermediate structure.

For a CTC input sequence of length $\{x_{1,}...,x_T\}$, CTC first computes a $N+1$ dimension vector at each time step. N is the number of elements in the vocabulary V. Then at every time step i, CTC is mapping this output vector to the output the distribution $\vec{\mathrm{p}}_{\mathrm{i}} = \{\mathrm{p}_{\mathrm{i},1},...,\mathrm{p}_{\mathrm{i,N+1}}\}$ by the SoftMax operation. here, $\mathrm{p}_{\mathrm{i}\,j}(j < N+1)$ is the output the probability of the j-Th element in time i, and p_i is the output the output the probability of the gap label '-'.

Once the computation is complete, CTC mapping its input sequence $\{x_{1,}...,x_T\}$ to a probability sequence $\mathrm{p} = \{\vec{\mathrm{p}}_1,...,\vec{\mathrm{p}}_T\}$ of the same length.

In case we select the w_i member from the set $V + \{\text{'-'}\}$ and put them in chronological order, what we obtain is an output sequence P = { w_1, ..., W_T} of the length T. P is a path. That' s the path definition. Since p_i is the probability of the output of the w_i element of $V + \{\text{'-'}\}$ at moment i, the probability of the path P can be calculated as equation (13).

$$\mathrm{p}(P) = \prod_{i=1}^{T} p_{i,w_i} \quad . \tag{13}$$

The above is the process of calculating the path probability. In fact, the length of the transcription is much shorter than the input sequence, and the process in which the path is of the same length T as the input sequence does not match the actual situation. Therefore, we merge a number of related paths into a shorter sequence of tags. This is the path merging process. It consists of two main operations:

Remove duplicate labels. In case there are multiple identical outputs in consecutive time steps, only one of them is kept. For example, for two different seven-step paths 'd-oo-g-' and 'dd-o-g-', the path after removing duplicate tags is both 'd-o-g-'.

Remove the blank tag '-' from the path. The '-' indicates that there is no output for this frame and it should be removed to get the final tag sequence. After removing all blank tags, the sequence d-o-g- becomes 'dog'.

In the merge process shown above, 'd-oo-g-' and 'dd-o-g-' are two length 7 paths, but the label sequence for 'dog' is length 3. For instance, supposing that the tag sequence 'dog' comes from a path of length 4, it contains 7 different paths as shown in Fig. 4.

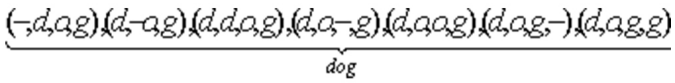

Fig. 4. The path length of the tag sequence 'dog' is 4

The fence grid of these paths is shown in Fig. 5. In this Fig. 1 to Fig. 4 represent the time steps and '-', 'd', 'o', and 'g ' represent the output of every time step. By moving in the direction of the arrows, each path that starts in time step 1 and stops on time step 4 is a legal path for the tag sequence 'dog'. Thus, a short label sequence can be merged by several long paths.

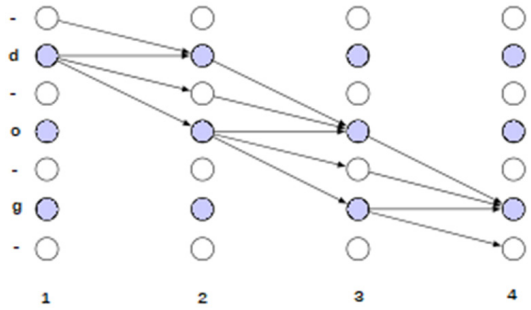

Fig. 5. Decode tag for 'dog' in path of length 4

Besides obtaining the final label sequence from the paths, in addition to the path merging process aims to calculate the probability of the final label sequence. For a tag sequence L composed of k paths $\{P_1, ..., P_k\}$ its probability $P(L)$ is calculated as shown in Equation (14):

$$P(L) = \sum_{i=1}^{k} p(P_i) \ . \tag{14}$$

As can be seen from the calculations described above, the probability of the tag sequence is differentiable. Therefore, we can use the back propagation algorithm to train the model to maximize the probability of the true tag sequence, and use the trained model to identify the speech with the maximum probability of the tag sequence as the final result.

2.5 Max Pooling

Now let's look at how CNN calculates t_c and window_c using the maximum pooling layer. We call the time span covered by the result element after CNN and the maximum pooling as t_p, and the time offset window between two adjacent elements as $\mathrm{window}_\mathrm{p}$. For the maximum $\mathrm{w}_\mathrm{p} \times h_p$ pool, the pool step size is $\mathrm{sw}_\mathrm{p} \times sh_p$, t_p and $\mathrm{window}_\mathrm{p}$ can be calculated according to t_c and window_c, as shown in Equation (15) :

$$\begin{aligned} \mathrm{t}_p &= \mathrm{t}_c + window_c \cdot (w_p - 1) \\ window_p &= sw_p \cdot window_c \end{aligned} \ . \tag{15}$$

Substituting Equation (2) into Equation (15), Equations (16) and (17) can be obtained:

$$\mathrm{t}_p = t_i + (w_c - 1) \cdot window_i + sw_c \cdot window_i \cdot (w_p - 1) \tag{16}$$

$$window_p = sw_p \cdot sw_c \cdot window_i \cdot \tag{17}$$

Equations (15) and (16) show that the largest pool can also expand the corresponding time span of the feature and reduce the calculation steps. Since max-pooling uses the maximum value as the output, it is helpful to select the most useful feature from many features with general functions, reduce the amount of calculation and the occurrence of over-fitting of the model.

2.6 Ernie

RNIE [20-21] is a knowledge - enhanced language representation model proposed by Baidu Inc. Inspired by the masking strategy of Bert (Bidirectional Encoder Representation from Transformers) [22-23], ERNIE was designed to learn language representations enhanced by knowledge masking strategies, which include entity layer masking and phrase layer masking. Entity-level policies mask entities that are usually composed of multiple words. Phrase-level strategies mask an overall conceptual unit of a few words. In this way, priori knowledge of phrases and entities is implicitly learned during training. ERNIE did not add knowledge embedding directly, but implicitly learned information about knowledge and longer semantic dependencies, such as the relationship between entities, the attributes of entities, the type of events, etc. To guide word embedding learning. This makes the model have better generalization and adaptability.

ERNIE model is an enhanced model based on knowledge masking strategy. By masking semantic units such as words and entities, the model learns the semantic representation of the complete concept. In terms of structure, the ERNIE model is mainly divided into Transformer [24], encoding and knowledge integration. The former uses Transformer as the basic encoder of the model to generate the corresponding word vector representation to retain the context information of the word in the text. The latter integrates phrase and entity-level knowledge into language representation through a multi-stage knowledge masking strategy. The ERNIE model structure is shown in Fig. 6.

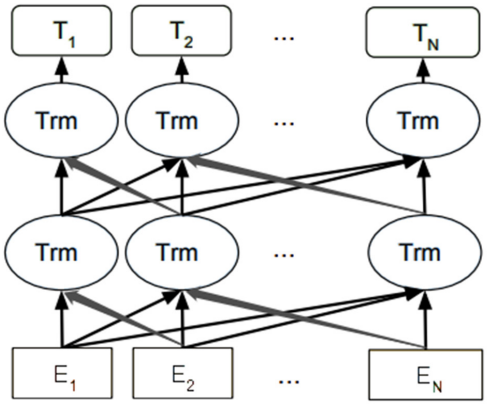

Fig. 6. Schematic diagram of ERNIE model structure

As can be seen from the structure in Figure 6, the output part of the model contains the word vector representation of the text context information, and each word vector $[T_1, T_2, ...T_n]$ contains the text information of the whole sequence. Due to the language of the traditional model in forecasting the next word for training target, using two-way code will make the words need to predict indirectly between multi-level context "see", that is to say, if I want to predict the input E_n at time t, each input will see in hidden layer targets information of E_n, leads to information leakage. To solve this problem, the Bert model turns the input at the corresponding position into a [mask] mark, masking a portion of the input sequence at random. ERNIE's model is further optimized on this basis, and a multi-stage knowledge masking strategy is proposed, which increases the masking of words to the masking of phrases and entities. ERNIE model further enhances the semantic representation of complete concepts in sentences by learning the knowledge of entity concepts. To achieve this, ERNIE uses a multi-stage knowledge masking strategy. Fig. 7 is divided into three stages:

Sentence	Harry	Potter	is	a	series	of	fantasy	novels	written	by	British	author	J.	K.	Rowling
Basic-level Masking	[mask]	Potter	is	a	series	[mask]	fantasy	novels	[mask]	by	British	author	J.	[mask]	Rowling
Entity-level Masking	Harry	Potter	is	a	series	[mask]	fantasy	novels	[mask]	by	British	author	[mask]	[mask]	[mask]
Phrase-level Masking	Harry	Potter	is	[mask]	[mask]	[mask]	fantasy	novels	[mask]	by	British	author	[mask]	[mask]	[mask]

Fig. 7. Different levels of masking of sentences

The first stage is basic masking, which takes the whole sentence as a sequence of basic language units and masks it by word. However, this way is random masking at the word level, so it is difficult to complete modeling of high-level semantics.

The second stage is phrase-level masking, where a phrase is used as the basic masking unit. To randomly select the sentence phrase, all words in the same phrase are turned into [mask] markers and predicted, so that the phrase information can be retained well in the embedding process.

The third stage is entity level masking, where people, places, organizations, products, etc. Can be represented with appropriate names for entity information. In this stage, the named entity components in the sentence are first analyzed, then the entities in the sentence are randomly selected, and each word in the sentence is [mask] marked and predicted.

After three stages of processing, a word representation form composed of rich semantic information can be obtained finally. The relevance of each component in the sentence can be well preserved, and the semantic information of the important components will not be lost while the complete important information is obscured.

3 Experiment and Analysis

3.1 Experimental Environment

This experiment is based on the PaddlePaddle platform. The specific configuration and experimental environment are shown in Table 1.

Table 1. Experimental environment and configuration

Experimental Environment	Experimental Configuration
CPU	4 Cores. RAM: 32GB. Disk: 100GB
GPU	Tesla V100. Video Mem: 16GB
Programming Language	Python 3. 6
Deep Learning Framework	PaddlePaddle 1. 8. 0

3.2 Experimental Data Set

The ERNIE-BiLSTM model was verified experimentally using the very classic NLPCC2014 data set. This data set is the sample data of Weibo sentiment analysis, which contains 15159 pieces of data, including 7650 pieces of positive data and 7509 pieces of negative data, both in UFT-8 coding format. The data are divided into training set data (train. tsv), validation set data (dev. tsv), test set data (test. tsv), and to be inferied data(infer. tsv). The data used for training, prediction and evaluation are shown in Table 2.

Table 2. Empirical data

label	text_a
1	书 的 质量 和 印刷 都 不错 字 的 大小 也 刚刚 好 很 清楚 喜欢
0	赶快 进货 啊 而且 卓越 书 太贵 打 的 折扣 太低 了
0	音质 画面 都 不怎么 好 大概 是 老片 的 缘故
1	界面 美观 使用方便 扫描 速度 快 上网 安心 了

We trained our model entirely on the Aishell-1 [25] corpus, with database voice data for 178 hours and 400 speakers. Corpus is divided into training set, verification set and test set. The training set consisted of 120, 098 utterances from 340 people, the verification set consisted of 14, 326 utterances from 40 people, and the test set consisted of 7, 176 utterances from 20 people. Each speaker recorded about 360 utterances (approximately 26 minutes of utterances). Table 3 provides a summary of all subsets in the corpus.

Table 3. Data set

Date Set	Duration (hours)	Number of Men	Number of Women
Training Set	150	161	179
Validation Set	10	12	28
Testing Set	5	13	7

The MFCC is used as the input feature of our model. This involves the 13-dimensional MFCC and the first-and second-order difference coefficients (39-dimensional features in total.) MFCC characteristics are derived from the original audio file with a frame window range of 25 ms and a shift window of 10 ms between consecutive frames.

The decoding target contains 4334 characters in the AISHELL-1 database audio transcription, but the input of CTC is 4335 dimensions, which is due to the blank label "-" is supposed to be inserted in the vocabulary during CTC decoding.

3.3 Analysis of Experimental Parameters and Model Training Results

The basic parameters of the ERNIE-BiLSTM model are shown in Table 4, with a total of 12 network layers and a 12-head model. During the training process, the Adma optimizer is used and the learning efficiency is chosen as 0. 00002.

Table 4. Parameter setting of experiment part

Parameter	Set Point	Parameter	Set Point
batch_size	32	num_hidden_layers	12
epoch	10	BiLSTM_layers	4
hidden_size	768	max_seq_len	512
num_attention_heads	12	num_labels	2

In order to accurately and intuitively reflect the quality of each model, the program is set to use verification set for testing after each epoch. The performance changes of each model after each epoch were observed through the change of accuracy and loss values of the verification set. The test results are shown in Table 5.

Table 5. Comparison results of model tests

Model	Accuracy (%)	F1 value	Recall (%)
RNN	80. 26	0. 7980	79. 31
BiLSTM	83. 71	0. 8404	84. 57
ERNIE	90. 48	0. 9042	90. 47
ERNIE-BiLSTM	96. 35	0. 9618	96. 02

Fig. 8 shows the change of accuracy rate V of the verified set of the deep learning model, and the change of loss value is shown in Fig. 9.

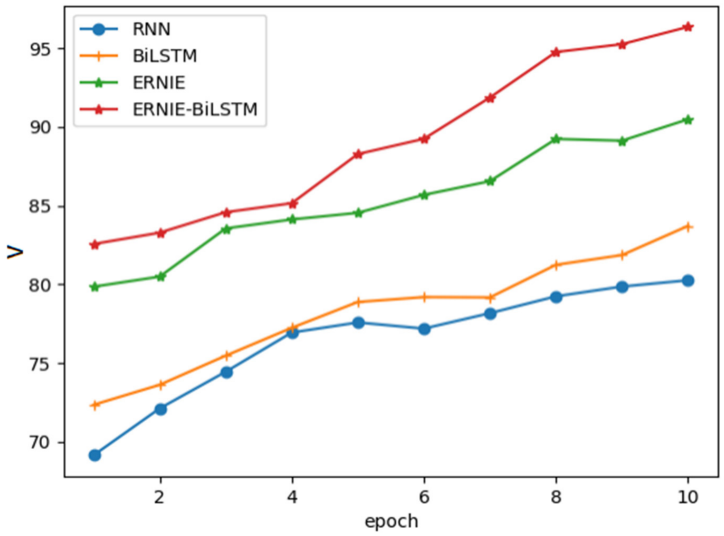

Fig. 8. The variation curve of the accuracy V

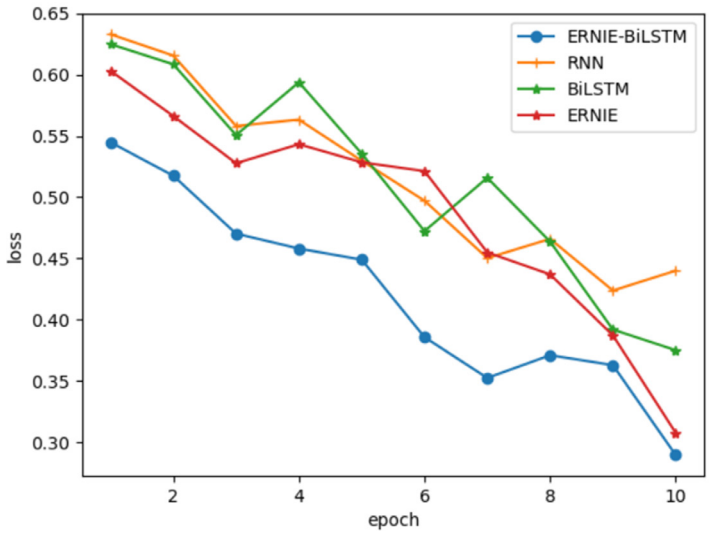

Fig. 9. Variation curve of loss value

The number of hidden units in CNN-BiGRU-CTC model has a great influence on the performance of BiGRU model. The different number of hidden units means that BiGRU models the context information and the current input in different dimensions. Excessive dimensions may introduce unnecessary features, which confuses the recognition model and leads to overfitting and reduces the recognition accuracy. At the same time, it doesn't work properly in lower dimensions. Therefore, to select the optimal number of hidden units, the performance of models with different dimensions is shown in Table 6 below. We increased the hidden dimension from 256 to 1024 for experiment, and the model performance improved continuously. However, when the dimension increased to 1024, the performance began to decline. Therefore, when the hidden dimension is 768, the model performance reaches the best 18. 7%. Therefore, we believe that this is because 1024 dimension introduces unnecessary features, which affects the performance of the recognition model. As for the model, the dimension is too high, which reduces the WER result.

BiGRU allows modeling of contextual information from two directions. In this paper, by comparing the model performance with different BiGRU depths, we show the results in Table 7. The model with two BiGRU layers has worse performance than the model with only one BiGRU layer. Therefore, using more BiGRUs only involves

more useless features and unnecessarily expands the context, thus confusing the recognition model and degrading the performance.

Table 6. Comparison of different hidden element number models in BiGRU

Hide Layer Dimensions	WER (%)
256	23. 5
512	20. 4
768	18. 7
1024	21. 5

Table 7. Comparison of different BiGRU depth models

BiGRU Layer	WER (%)
1	18. 7
2	24. 3

From Table 8, it can be seen that different depths of CNN layers and the presence or absence of BiGRU layers have a significant impact on the speech recognition accuracy. Each CNN layer has 64 feature maps.

Table 8. Experimental results of the number of CNN layers and the presence or absence of BiGRU layers

CNN Depth	WER（%）	
	Without BiGRU	With BiGRU
2	64.8	25.4
3	61.2	23.2
4	56.9	30.7
5	69.8	36.8

It can be seen that when there are three layers of CNN and BiGRU layer, the model has the best effect on speech recognition. For the model without BiGRU, the CNN layer increased from 2 layers to 5 layers, but the recognition effect is not better with the increase of the number of layers, because the gradient may disappear if the CNN layer is overlaid excessively, which leads to the degradation of the model performance. Because Bi-GRU can model the context information, there are only three layers of CNN instead of four in terms of performance from increasing to reducing early arrival.

3.4 Analysis of Experimental Results

In order to simulate the actual situation, CASIA Chinese emotion corpus will be used, which is recorded by the Institute of automation, Chinese Academy of Sciences and consists of four professional speakers with 9600 sentences of different pronunciation. Take 800 sentences of positive emotion and 800 sentences of negative emotion. As we all know, in speech recognition, the recognized Chinese text data will appear individual wrong words, wrong words. After the text data is transferred into Ernie language pre training model, the model can still accurately classify the text emotion, with good classification accuracy. The conclusion can be confirmed by the following Table 9.

Table 9. Comparison results of different models

Model	Accuracy (%)	F1 value	Recall (%)
CNN-BiGRU-CTC+ERNIE-BiLSTM	94. 50	0.9456	94. 62
CNN-BiGRU-CTC+BiLSTM+Word2vec	82.32	0.8337	83.68

3.5 Model Summary

Compared with other models [26] that directly use voice signals for sentiment analysis, the advantage of the model in this paper lies in the accurate acquisition of users' current emotions while obtaining users' voice information, the model can be applied to intelligent customer service scenarios. Our CNN-BiGRU-CTC + ERNIE-BiLSTM model achieves 94. 5% classification accuracy of sentiment polarity which is 5.24% better than the latest speech sentiment analysis model. The reason for this good classification accuracy is that the ERNIE-BiLSTM model can still have good classification accuracy and good robustness in case of individual spelling and pronunciation errors in speech recognition. And, unlike traditional speech sentiment models, the model in this paper has access to textual data information and can obtain sentiment information in semantics.Good progress has been made, but there are still some shortcomings that need to be improved. For example, translating speech into text will lose some data information, such as voice tone and volume level. In this paper, multiple scenes are used for data transmission, so there is a need to further improve multiple scenes into multiple modes. The combination of speech and text information and the use of specific algorithms for data integration, rating and classification of speech and text still needs further research.

4 Conclusions

In this paper, a new model CNN-BiGRU-CTC + ERNIE-BiLSTM is proposed for multi-scene sentiment analysis of speech and text. An end-to-end ASR system is used to implicitly learn a language model with adequate speech transcription. The speech signal is fed into the model to finally obtain the emotional polarity and semantic information of the speech. Our model achieves 94.5% accuracy in classifying sentiment polarity. The model can be applied in intelligent customer service systems to improve the accuracy, comfort and user experience of message replies. It is experimentally proven that the model has better stability and practicality as well as innovative new ideas compared to the traditional models of speech recognition. Also, since the corpus we use is freely available, our model is reproducible and comparable, which provides a new baseline for further research on Chinese ASR sentiment analysis.

References

[1] A.B. Ingale, D.S. Chaudhari, Speech emotion recognition, International Journal of Soft Computing and Engineering (IJSCE) 2(1)(2012) 235-238.
[2] G.N. Peerzade, R.R. Deshmukh, S.D. Waghmare, A review: Speech emotion recognition, International Journal Comput. Sci. Eng. 6(3)(2018) 400-402.
[3] M. Swain, A. Routray, P. Kabisatpathy, Databases, features and classifiers for speech emotion recognition: a review, International Journal of Speech Technology 21(1)(2018) 93-120.
[4] R.A. Khalil, E. Jones, M.I. Babar, T. Jan, M.H. Zafar, T. Alhussain, Speech emotion recognition using deep learning techniques: A review, IEEE Access 7(2019) 117327-117345.
[5] A. Hannun, C. Case, J. Casper, B. Catanzaro, G. Diamos, E. Elsen, R. Prenger, S. Satheesh, S. Sengupta, A. Coates, A. Y. Ng, Deep speech: Scaling up end-to-end speech recognition (2014) arXiv preprint arXiv:1412.5567.
[6] A. Graves, N. Jaitly, Towards end-to-end speech recognition with recurrent neural networks, in: Proc. International conference on machine learning. PMLR, 2014.
[7] G. Trigeorgis, F. Ringeval, R. Brueckner, E. Marchi, M. A. Nicolaou, B. Schuller, S. Zafeiriou, Adieu features? end-to-end speech emotion recognition using a deep convolutional recurrent network, in: Proc. 2016 IEEE international conference on acoustics, speech and signal processing (ICASSP). IEEE, 2016.
[8] S. Latif, R. Rana, S. Khalifa, R. Jurdak, J. Epps, Direct modelling of speech emotion from raw speech (2019) arXiv pre-

print arXiv:1904.03833.
[9] P.C. Li, Y. Song, I.V. McLoughlin, W.D. Guo, R. Li, An attention pooling based representation learning method for speech emotion recognition, 2018.
[10]W. Lim, D. Jang, T. Lee, Speech emotion recognition using convolutional and recurrent neural networks, in: Proc. 2016 Asia-Pacific signal and information processing association annual summit and conference (APSIPA). IEEE, 2016.
[11]Y. Zhang, B. Wallace, A sensitivity analysis of (and practitioners' guide to) convolutional neural networks for sentence classification (2015) arXiv preprint arXiv:1510.03820.
[12]Y. Kim, Y. Jernite, D. Sontag, A. M. Rush, Character-aware neural language models, in: Proc. Thirtieth AAAI conference on artificial intelligence, 2016.
[13]J. Murphy, An overview of convolutional neural network architectures for deep learning, Microway Inc, 2016.
[14]O. Abdel-Hamid, A. Mohamed, H. Jiang, D. Li, G. Penn, D. Yu, Convolutional neural networks for speech recognition, IEEE/ACM Transactions on audio, speech, and language processing 22(10)(2014) 1533-1545.
[15]S. Ioffe, C. Szegedy, Batch normalization: Accelerating deep network training by reducing internal covariate shift, in: Proc. International conference on machine learning. PMLR, 2015.
[16]X. Glorot, A. Bordes, Y. Bengio, Deep sparse rectifier neural networks, in: Proceedings of the fourteenth international conference on artificial intelligence and statistics. JMLR Workshop and Conference Proceedings, 2011.
[17]A. Graves, S. Fernández, F. Gomez, J. Schmidhuber, Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks, in: Proceedings of the 23rd international conference on Machine learning, 2006.
[18]S. Hochreiter, J. Schmidhuber, Long short-term memory, Neural computation 9(8)(1997) 1735-1780.
[19]H. Lee; W. H. Kang; S. J. Cheon; H. Kim; N. S. Kim, Gated Recurrent Context: Softmax-Free Attention for Online Encoder-Decoder Speech Recognition, IEEE/ACM Transactions on Audio, Speech, and Language Processing 29(2021) 710-719.
[20]Y. Sun, S. Wang, Y. Li, S. Feng, X. Chen, H. Zhang, X. Tian, D. X. Zhu, H. Tian, H. Wu, Ernie: Enhanced representation through knowledge integration (2019) arXiv preprint arXiv:1904.09223.
[21]Y. Sun, S. Wang, Y. Li, S. Feng, X. Chen, H. Tian, H. Wu,H. F. Wang, Ernie 2.0: A continual pre-training framework for language understanding, in: Proceedings of the AAAI Conference on Artificial Intelligence, 2020.
[22]Y.M. Cui, W.X. Che, T. Liu, B. Qin, Z.Q. Yang, S.J. Wang, G.P. Hu, Pre-training with whole word masking for chinese bert (2019) arXiv preprint arXiv:1906.08101.
[23]A. Wang, A. Singh, J. Michael, F. Hill, O. Levy, S. R. Bowman, GLUE: A multi-task benchmark and analysis platform for natural language understanding (2018) arXiv preprint arXiv:1804.07461.
[24]A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A.N. Gomez, L. Kaiser, Attention is all you need, in: Proc. Advances in neural information processing systems, 2017.
[25]H. Bu, J. Du, X. Na, B. Wu, H. Zheng, Aishell-1: An open-source mandarin speech corpus and a speech recognition baseline, in: Proc. 2017 20th Conference of the Oriental Chapter of the International Coordinating Committee on Speech Databases and Speech I/O Systems and Assessment (O-COCOSDA). IEEE, 2017.
[26]B. Salian, O. Narvade, R. Tambewagh, S. Bharne, Speech Emotion Recognition using Time Distributed CNN and LSTM, in: Proc. ITM Web of Conferences. EDP Sciences, 2021.