IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received November 6, 2018, accepted November 23, 2018, date of publication December 17, 2018,
date of current version February 4, 2019.

Digital Object Identifier 10.1109/ACCESS.2018.2887093

Chinese Dialogue Intention Classification Based on Multi-Model Ensemble

MANSHU TU, BING WANG, AND XUEMIN ZHAO

Key Laboratory of Speech Acoustics and Content Understanding, Institute of Acoustics, Chinese Academy of Sciences, Beijing 100049, China

Corresponding author: Manshu Tu (tumanshu_ucas@163.com)

This work was supported in part by the National Natural Science Foundation of China under Grant 11590770-4, Grant 61650202, Grant 11722437, Grant U1536117, Grant 61671442, Grant 11674352, Grant 11504406, and Grant 61601453, in part by the National Key Research and Development Program under Grant 2016YFB0801203, Grant 2016YFC0800503, and Grant 2017YFB1002803, in part by the Key Science and Technology Project of the Xinjiang Uygur Autonomous Region under Grant 2016A03007-1, and in part by the Foundation of Science and Technology on Information Assurance Laboratory under Grant KJ-17-102.

ABSTRACT In dialogue systems, understanding the user utterances is crucial for providing appropriate responses. A traditional dialogue act classification (DA) task is to classify each user reply into ''ACCEPT, REJECT, PROPOSE, and others''. In contrast, in this paper, we define the DA task on multiple round conversations between humans. The re-defined task is to classify a full dialogue according to the intention of one participant. We term this task as intention classification (IC). We, then, propose a hybrid neural network-based ensemble model for solving this problem. Two novel ensemble schemes are introduced for combining the classification results or features from various classifiers. One is ensembling features from each individual classifier using stacking, and we term this scheme as SFE. The other is adding wrong examples' weight to loss functions of each individual classifier using the AdaBoost scheme, and we term this scheme as MN-Ada. We have empirically examined the performance of the proposed ensemble schemes by using three popular deep neural networks, as well as one newly modified networks for IC. Extensive experiments have been conducted on a Chinese dialogue corpus. Our model can achieve state-of-the-art accuracy on the experimental dialogue corpus.

INDEX TERMS Dialogue intention classification, ensemble schemes, CNN, LSTM.

I. INTRODUCTION

Traditional DA task is to assist in generating smooth human-machine conversations. Therefore, it focuses on the task of classifying the responses of the human participant regarding the machine's suggestion. DA task obtain a category for each sentence. In this paper, we are interested in a more complicated dialogue scenario. For telecom carriers, custom services receive a huge amount of calls regarding charges, package enquiry and so on. Monitoring and analysis such calls can assist the companies to figure out custom preferences, their own service quality, and potential operating problems. In such a scenario, the specific task is to figure out the real intention of customers from a full dialogue. For example, costumers may call to enquire about some usage of their charges or change a phone package. We need to classify the purposes of these calls. This task is termed as intention classification (IC) task. Part of the challenges lie in multiple round conversations, spoken language and imperfect speech recognition results (Dialogues are the output of a speech recognition module). Besides, in numerous cases, the caller changes their minds over the conversation. Some examples of such conversations is given in Appendix A. It is worth mentioning that our system is already on line, we can deal one million dialogues one day.

IC task in essence is a special case of document classification (DC). In nature language processing (NLP), DC is one of the fundamental tasks. It has broad applications including topic classification [1], sentiment analysis [2] and information retrieval like spam detection [3]. Traditional approaches for DC are based on support vector machine (SVM) with hand-crafted features like Term Frequency–inverse Document Frequency (TF-IDF) feature [4]. Recently, deep neural networks are wildly used for learning better representation in DC. Related studies include Convolutional neural networks (CNN) [5] and Long Short-Term Memory networks (LSTM) based on Recurrent Neural Networks (RNN) [6].

In real world DC applications, instead of relying on individual machine learning classifiers, in order to achieve better prediction performance, researchers often apply ensemble to classifiers [7], [8]. Ensemble methods are learning algorithms that construct a set of classifiers. Predictions are made by

taking a vote from each individual classifiers [9]. Commonly used ensemble functions include bagging [10]–[12], boosting [13] and stacking [14]. Bagging is a method that generate multiple versions of a predictor and use them to get an aggregated predictor. It is often stuck in over-fitting when the number of ensemble models is relatively small. Stacking is a method similar to K-fold cross-validation, which can alleviate the over-fitting problems. It is a hierarchical structure, the output of one layer is the input of its next layer. Traditional stacking only ensembles outputs of each predictor as inputs to its next layer, we abbreviate this traditional method as SRE.

Given the real word application context of IC, in this paper, we focus on ensemble schemes that can take advantages of the outputs of multiple classifiers. SRE only considers the classification result of each involved classifiers. As mentioned above, when diverse features are included in classification tasks, predication accuracy usually increase. Inspired by this, instead of a single classification result for stacking, we propose to ensemble abstract features trained from each involved classifier for stacking (SFE). In SFE, we splice together abstract features extracted from each classifier for preserving more information, and ensemble them as input of a new classifier for final predictions.

For boosting schemes, traditional boost uses machine learning methods as weak classifiers and train weak classifier one by one. In order to train each individual weak classifier to focus on different aspects of data features, we need to hand picking features as input. We adapt the boost scheme to our IC task by incorporating weight information to the loss function of each individual classifier so that the above training process can automatically adjust training focus.

In order to test the proposed schemes, we select three popular NN based classifiers. Since IC in essence is a DC task, the selected classifier are reported to show competitive performances in normal DC tasks. However, in IC tasks participants might change minds during conversations. To capture this dynamic, we modified a CNN-LSTM so that the new classifier can capture inversion of intention. We use multi channel CNN with bidirectional LSTM (MCNN-BLSTM) to capture more features about each sentence in a dialogue. Adding reverse inter-sentence information to make the neural network capture the real intention proposed at the beginning of a dialogue. Our contributions are summarized as follows:

1) We propose a new task about dialogue intention classification. It classifies a full dialogue into a category, which can represent the final intention of a participant.
2) For individual classifier of the IC task, we modify CNN-LSTM by taking into consideration that participants' intention might change in a dialogue.
3) We propose two multi-model ensemble schemes. Experimental results show that our ensemble schemes can achieve state-of-the-art result.

The rest of this paper is organized as follows: Section 2 presents the related work. The details of MCNN-BLSTM are given in Section 3. The two ensemble scheme we proposed are given in Section 4. Experimental results and discussion are presented in Section 5. We draw conclusions in Section 6.

II. RELATED WORK

Classifier ensembles have shown promising advantages over single classifiers. Many experiments have verified this view. Drucker compares the performance of three types of neural network-based ensemble techniques to that of a single neural network. He finds that for a given computational cost, boosting is always best [15]. Maclin evaluates Bagging and Boosting on 23 datasets using both neural networks and decision trees as classification algorithm [16]. The results show that an ensemble is often more accurate than any of the single classifiers in the ensemble.

Then, some more effective ensemble schemes based on traditional bagging and boosting are proposed. For example, random forest (RF) adds additional randomness to bagging [17]. This counterintuitive strategy turns out to perform better than individual classifiers. It is also robust against over-fitting. These two methods show best accuracy on some synthetic datasets. AdaBoost for multi-class classification is proposed by Zhu *et al.* [18]. It directly extends the AdaBoost algorithm to the multi-class case without reducing it to multiple two-class problems.

Recent years, some researchers put their attentions on using ensemble methods with deep neural networks. Ju investigated multiple widely used ensemble methods with deep neural networks as candidate algorithms in image recognition tasks [19]. Across all of their experiments, the Stacking achieved best performance among all the ensemble methods in image classification. Strauss propose to use ensemble methods as a defense strategy against adversarial perturbations in deep neural networks [20]. The experiments show that ensemble methods not only improve the accuracy of neural networks on test data but also increase their robustness against adversarial perturbations. According to these works mentioned above of ensemble functions, we proposed our new ensemble schemes based on Adaboost and Stacking, which are proven effective. We use neural networks as basis model for they are exploited in DC task to automatically learn a good representation.

For DC problems, Common solutions are CNN, LSTM, their variants or combinations [3]. Kim [21] and others first propose to use CNN as a text classifier. In their study, the authors test different parameters and unsupervised pre-training of word vectors with fine-tuning to obtain the best classification result. Yin propose a multichannel variable-size Convolution (MVCNN) for DC tasks. MCVNN combines diverse versions of pre-trained word embeddings and use variable-size convolution filters to extracts features [22]. It achieves state-of-the-art performance on four tasks including small-scale binary, small-scale multi-class and large-scale Twitter sentiment prediction and subjectivity classification. Ji and Dernoncourt present a model based on RNN and CNN that incorporates preceding short text [23]. They also conduct a detailed comparison on the classification effect of RNN

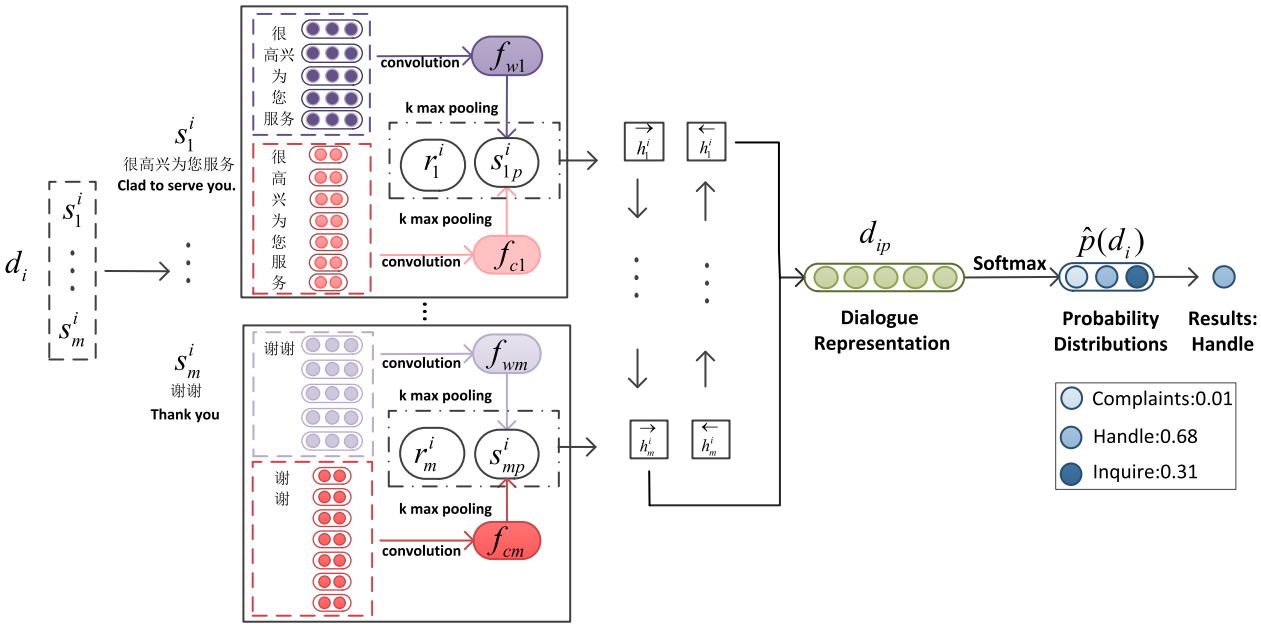


FIGURE 1. The framework of the MCNN-BLSTM model. Take the appendix A-A as example, we show the word-level input and character-level input for the first and last sentences, respectively. We also show the padding function for MCNN-LSTM in the figure for the last sentence. For this example, the final category is Handle because of it obtain the biggest score in the probability distribution.

and CNN for precessing short text. In different datasets the two classifiers have different performance, but both better than linear classifier like Support Vector Machine (SVM), naive Bayes. The model C-LSTM combines CNN and LSTM for sentence representation [24]. It utilizes a CNN to extract a sequence of higher-level phrase representations, and then feed them into LSTM to obtain the sentence representation.

III. MCNN-BLSTM

The architecture of this model is shown in Figure 1. Assume that a dialogue has m sentences and each sentence contains n words and t characters. There are total D dialogues in datasets. The proposed model first projects a full dialogue into a vector representation, on which we bulid a classifier to perform IC. In the following, we present how we bulid the dialogue level vector progressively from word vectors by using the hierarchical structure of MCNN-BLSTM.

A. SENTENCE REPRESENTATION

We denote a sentence with $s_j^i = \{r_j, w_{j1}^i/c_{j1}^i, w_{j2}^i/c_{j2}^i \cdots w_{jn}^i/c_{jt}^i\}$, i refers to the ith dialogue in a dataset, r is the role information, w_{jn}^i and c_{jt}^i are the words and the characters of s_j^i. We first embed them into two vectors. One is a word embedding e_{sw_j}, the other is a character embedding e_{sc_j}. Then, a convolutional step is carried out to convert these two embeddings to their corresponding feature maps. For the word embedding e_{sw_j}, its feature map fw_j is calculated by $fw_j = \phi(\omega_1 \cdot e_{sw_j} + b_1)$. For the character embedding e_{sc_j}, its feature map fc_j is calculated by $fc_j = \phi(\omega_2 \cdot e_{sc_j} + b_2)$. ω_1 and ω_2 are both a set of different convolution kernels in the MCNN and shared by every sentence.

Then we take a $k\_max\_pooling$ in fw_j and fc_j. Top k values in both fw_j and fc_j are extracted to form two fixed length vectorial representations, respectively. The two vectorial representations are concatenated to get an annotation of the sentence s_j^i, as demonstrated in the following equation.

$$s_{jp}^i = [k\_max\_pool(fw_j), k\_max\_pool(fc_j)] \tag{1}$$

B. DIALOGUE REPRESENTATION

After the processing above, a dialogue d_i can be converted to an ordered set $d_i = \{s_{1p}^i, s_{2p}^i \cdots s_{jp}^i \cdots s_{mp}^i\}$. Then, we use a bidirectional LSTM to obtain an annotation for d_i by summarizing information from two sequences constructed from d_i. The bidirectional LSTM contains a forward LSTM $\overrightarrow{f}$, which reads the dialogue d_i from s_{1p}^i to s_{mp}^i and a backward LSTM $\overleftarrow{f}$, which reads from s_{mp}^i to s_{1p}^i:

$$\overrightarrow{h_j^i} = \overrightarrow{LSTM}(s_{jp}^i) \quad j \in [1, m] \tag{2}$$

$$\overleftarrow{h_j^i} = \overleftarrow{LSTM}(s_{jp}^i) \quad j \in [m, 1] \tag{3}$$

We then generate the representation d_{ip} of d_i by concatenating the forward last output $\overrightarrow{h_m^i}$ and the backward last output $\overleftarrow{h_1^i}$, i.e. $d_{ip} = [\overrightarrow{h_m^i}, \overleftarrow{h_1^i}]$.

C. DIALOGUE CLASSIFICATION

The dialogue vector d_{ip} is a high level representation of d_i. We use it as input features for SFE. Then, d_{ip} is sent to a fully

connected layer and softmax for classification:

$$\hat{p}(d_i) = softmax(\omega_3 * d_{ip} + b_3) \tag{4}$$

where $\hat{p}(d_i)$ is the probability distribution of each intention category. We use cross entropy as training loss:

$$L_{mcnn\_lstm} = -\frac{1}{D}\sum_{i=1}^{D}[p(d_i)log^{\hat{p}(d_i)}] \tag{5}$$

where $p(d_i)$ is the real probability distributions of each category. The category with the highest probability is the final classification result of d_i.

D. DETAILS OF PADDING

In NLP, padding is used to compensate places at the edge of an input so that words at the beginning and the end of a sentence can be involved in convolution. The same idea applies to dialogues as well. For MCNN-BLSTM, we first apply padding to each sentence so that all of them are of the same length n or t. Also each dialogue goes through padding to get same length m. If any sentence or dialogue length is larger than predefined parameters, they are truncated.

Model-R: In our IC task, the dialogue participants have different roles, inquirer, custom service and irrelevant people. In order to involve in such information in the classification models, we extend the representation of a sentence s^i_{jp} to $s^i_{jp} = [r^i_j, s^i_{jp}]$, r^i_j refers to the role information of s^i_j. As shown in Figure 1, the r^i_1 and the r^i_m represent the role information of s^i_1 and s^i_m, respectively. Specifically, we use numerical values 1, 2, and 3 to represent these roles. In this article, we attach a letter "R" to a model name, like MCNN-BLSTM-R, to indicate that the role information is included in the sentence representation.

E. RELATION TO CNN-LSTM

The MCNN-BLSTM model is designed based on the model CNN-LSTM [25]. Different from the original model, we use both words and characters to represent sentences. We add characters' feature for making up some errors caused by word segmentation. We also use bidirectional LSTM instead of the original LSTM, because the real intention may appear at the beginning or the end of a dialogue. The further back a cell locates in LSTM, the more important it is. Using bidirectional LSTM can consider intentions at either the beginning or the back of a sentence.

IV. MULTI-MODEL ENSEMBLE SCHEME

In this section, we introduce our Multi-model ensemble scheme for the proposed IC task. We first introduce the scheme that use AdaBoost to neural networks. Then we present our improved stacking scheme SFE in details. Let $T = T_1, \cdots, T_b, \cdots, T_M$ denote individual classifiers in these two ensemble scheme, M is the total number of classifiers.

A. MN-ADA

Figure 2(a) presents the architecture of MN-Ada. Traditional AdaBoost is a series structure, which trains weak classifiers one by one. It pays more attention to misclassified dialogue by increasing the weighting μ_i at its next model. The initial weight of each dialogue is $\mu_i = 1/D$. We apply this idea to neural networks by adding μ_i in loss function. Each classifier T gets a weight α^T based on its performance:

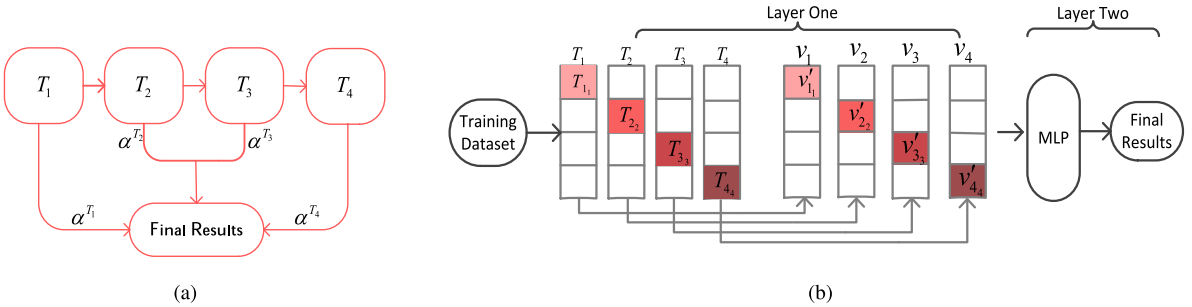
$$\alpha^T = log\frac{1 - err_T}{err_T} + log(K - 1) \tag{6}$$

where err_T is the error rate of individual classifier T. The K represents the number of categories. The bigger err_T is, the smaller α^T. It means the less effective the model is, the smaller the contribution to the final ensemble. The $log(K - 1)$ is to guarantee that the value of α^T is positive, when value of err_T is bigger than 1/2. After getting α^T, we update μ_i by Equation (8).

$$g(f(x)) = \begin{cases} 1, & f(x) = True \\ 0, & f(x) = False \end{cases} \tag{7}$$

$$\mu_i \leftarrow \mu_i \bullet exp(a^T \bullet g(f(x))),$$
$$i = 1, \cdots, m \quad f(x) : T(d_i) \neq k \tag{8}$$

where $T(d_i)$ is the classification result of dialogue d_i obtained from individual classifier T, k is the true label of d_i. The value



(a) (b)

FIGURE 2. An overview of ensemble scheme. Fig a presents the architecture of MN-Ada and Fig b is the architecture of SFE. In Figure 2(b), different colour represents different models. In this paper we use four models. The coloured square which contains character v_i represents features that the corresponding validation set (have same colour with character T) get from trained model T_i. (a) MN-Ada. (b) SFE.

of $g(f(x))$ can only be zero or one. When $f(x)$ is true, the value of $g(f(x))$ is one, otherwise it is zero. By the equation (8), when classifier T gives wrong result on d_i, μ_i becomes bigger in order to achieve greater weight distribution for misclassified dialogue. Then we add μ_i to cross entropy loss L for every classifier T, which is calculated by Equation (9).

$$L_T = -\frac{1}{m}\sum_{i=1}^{m}\mu_i[p(d_i)log^{\hat{p}(d_i)}] \tag{9}$$

The $p(d_i)$ and $\hat{p}(d_i)$ in loss function represent the actual probability distribution and the predicted probability distribution, respectively. By multiplying weights μ_i to the loss function, we make a current classifier place more emphasis on examples that a previous classifier gives wrong results. We then repeat these steps for each classifier and calculate a weight α^T for each individual classifier. We also get a result $\hat{k}$ from each individual classifier for each dialogue d_i. Then the final category is obtain by Equation (10).

$$final\_result(d_i) = argmax_{\hat{k}}(\sum_{T}^{M}\alpha^T \bullet g(f(x)))f(x) : T(d_i) = \hat{k} \tag{10}$$

The final predicted category of d_i is the value $\hat{k}$ that makes Equation (10) maximum.

B. SFE

Figure 2(b) presents the architecture of SFE. In this scheme, we first divide a training set into M different sets, which is the same as the number of models. Similar to a k-fold cross validation, $M-1$ out of the M folds are treated as a training set, the rest one set serves as a validation set. For each split of the training set and the validation set (M split in total), we train one individual classifier, we call this individual classifier a sub-classifier T_{b_r}, $b \in [1, M]$, $r \in [1, M]$.

Next, for the ensemble step, we apply another classifier, which is a multilayer perceptron (MLP). For the input of this MLP, in traditional stacking scheme, it is a vector constructed from the output (i.e. predicted category of each dialogue) of each individual classifier. In our proposed ensemble scheme, we use the input of each individual classifier's last hidden layer to construct the input. Obviously, the new input vector is much larger than the traditional one. We hypothesise that such an operation can preserve more information from each individual classifier for the ensemble to perform classification, and it can further improve classification accuracy.

Since this new input vector consists of component vectors provided by each individual classifier, we present how one component input vector is constructed, and the rest component input vector are generated in the same way. For one instance in the original training set, for each sub-classifier T_b we first decide which sub-classifier's validation set that the instance belongs to. Then, we propagate the instance through that specific sub-classifier to its last hidden layer to get a corresponding component vector v_{b_r}. When we ensemble a set of different sub-classifiers, the size of vectors v_b might be different. In order to convert each component vector to the same size, we then apply principal component analysis (PCA) to v_b, and only keep the first a values to get v'_{b_r}. Their process is applied to all M classifier to get a complete input matrix F for the MLP. The final classification is obtained by equation (12).

$$F = \begin{bmatrix} v'_{1_1} & \cdots & v'_{b_1} & \cdots & v'_{M_1} \\ v'_{1_2} & \cdots & v'_{b_2} & \cdots & v'_{M_2} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ v'_{1_M} & \cdots & v'_{b_M} & \cdots & v'_{M_M} \end{bmatrix} \tag{11}$$

$$final\_result = softmax(MLP(F)) \tag{12}$$

V. EXPERIMENT

In this section we introduce our datasets, data preprocessing, experiments setting and experiment results.

A. DATASETS AND DATA PREPROCESSING

We use two Chinese dialogue datasets. They are both provided by a telecommunication network carrier. They are speech recognition outputs of conversations between humans. As in Table 1, we can see that these two datasets both have the characteristics of short sentence length and long dialogue length. In our experiments, we set the max sentence, character and dialogue length 10, 15 and 22 in data1, and 12, 20, 26 in data2, respectively. The data1 has 40253 pieces of data and data2 has 15225 pieces of data, this amount of data is sufficient for training compare with the amount of data in the public data set like MR [1] and CR.[2] According to the contents of dialogues, we divide data1 into six categories and data2 into three categories. These categories are also defined in the on line system. Examples of classification criteria for data2 used in this study are given as follows:

1) Complaints: unsatisfied with the service.
2) Handle: handle a business.
3) Inquire: consult product features.

TABLE 1. This table presents statistics for two experimental datasets. First three columns represent the average length of words and characters in a sentence, and the average sentences number in a dialogue, respectively. Role Num represents number of roles in one datasets.

	Word Length	Character Length	Sentence Num	Role Num	Class Num	Total Num
Data1	8.02	11.00	21.32	None	6	40253
Data2	11.20	17.75	25.16	3	3	15225

We have conducted two series of experiments. In one experiment, the IC task is to classify given dialogues into three classes, which are complaint, handle and inquire. In the other experiment, the IC task is to distinguish whether the given dialogue is emotional or not. Data2 has three role categories, which means that there are up to three people

[1] https://www.cs.cornell.edu/people/pabo/movie-review-data/
[2] http://www.cs.uic.edu/~liub/FBS/sentiment-analysis.html

involved in one conversation. Data1 does not have role information. In our experiments, the ratio of the training set to the validation set for the two datasets is 9:1. We show three dialogue examples at Appendix A. They represent three different categories and they all have a intention change. For example, the example in Appendix A-C, although it has the intention of inquire at beginning, it belongs to the class complaints.

The preprocessing of the datasets includes changing traditional Chinese to simplified Chinese, converting Chinese number to Arabic numerals,[3] removing telephone number and ID number. We use ICTCLAS[4] to segment words. After obtaining the segmented dialogue, we generate a stop words dictionary based on our datasets by calculating TF-IDF value of each word and taking the top 5% from the smallest values to largest values.

B. IMPLEMENTATION DETAILS

For the word embedding process that generates vectorized presentation of words, 300-dimensional *word2vec* vectors are used for initialization. As for the char embedding, 50-dimensional *word2vec* vectors are used for initialization. We use the two experiment datasets and one more Sohu News Corps[5] to train them. The initial word embedding network weights are generated from range $[-0.01, 0.01]$ by using uniform distribution. All of the models are optimized with the Adaptive Moment Estimation(Adam) over shuffled mini-batches. We repeat each experiment for 30 times and each 10 times we use 10-flod cross-validation. All of the experiments are under Debian system, Linux 8.9 and python 3.6, tensorflow1.4.

For MCNN-LSTM, the filter width of ω_1 and ω_2 are $[1, 2, 3]$ and $[2, 3, 4]$ and the number of them are both 64. The value of *k_max_pool* is 8. For SFE, the *a* value we keep in PCA is 100.

C. PERFORMANCE COMPARISON

The baseline methods of individual classifier in experiments include:

1) CNN: It trained on top of pre-trained word vectors for our task [21]. It show that a simple CNN with little hyperparameter tuning and static vectors achieves excellent results on multiple benchmarks.
2) LSTM: It use LSTM to represent document feature and use softmax as classifier [26].
3) HAN: It has two levels of attention mechanisms applied at the word and sentence-level [27]. Enabling it to attend differentially to more and less important content when constructing the dialogue representation. We set the RNN size as 150, word attention size and sentence attention size as 100. The feature we want for SFE obtains after sentence attention operation, the feature size is 300.
4) VDCNN:The VDCNN model uses only small convolutions and pooling operations [28]. It achieves state-of-the-art results on a number of public classification datasets especially on short texts. We use a 9 layers architecture as it gets best performance in all experiments. Our input is characters of dialogues, the filter size is 3 and polling size is 3. The filter number for each convolutional block is 64, 128, 256, 512. The features for SFE are coming from the *k_max_pooling*, as described in the article the feature size is $1536 = 3 \times 512$.
5) BLSTM-2DCNN: This model is described in detail in [29]. It first uses BLSTM to encoder all embedded words in new representation. Then use two layer CNN to extract the full dialogue features from new representation. Then 2D max pooling operation is utilized to obtain a fixed length vector h^* and it is the whole representation of the input dialogue d_i. Then it passed to a 2 fully connected layers and then send to softmax classifier layer to predict the intention. The size of feature map in CNN both $3 * 3$, the 2D pooling size is $2 * 2$ and LSTM size is 300. The features for SFE are get after the first fully connected layer, the size is 128.

We chose the CNN and LSTM for baseline because of they are the most basic model for document classification. The other models we use for baseline because of they all get state-of-the-art accuracy on different document classification datasets.

The baseline methods of ensemble function in comparison include:

1) Random Forest(RF): Training four models separately using randomly selected data at same time. The probability of each categories in VDCNN is denoted as $P_{VDCNN} = p_1, p_2, \cdots, p_K$, and so on, we get $P_{VDCNN}, P_{HAN}, P_{MCNN-BLSTM}$, $P_{BLSTM-2DCNN}$. Different with traditional RF has same weights on each model, we assign different weights for them to get final result.

$$\begin{aligned} final\_result = {} & \alpha * P_{VDCNN} + \beta * P_{HAN} \\ & + \gamma * P_{MCNN-BLSTM} \\ & + \delta * P_{BLSTM-2DCNN} \\ & \alpha + \beta + \gamma + \delta = 1 \end{aligned}$$

2) SRE: Combing all results from four individual models and trained a supported vector machine (SVM) classifier to get final result.

Experimental results are given in the Table 2. The best results in corresponding columns are highlighted with bold font. ''Acc'' is an abbreviation of accuracy and ''Re'' is an abbreviation of recall. As explained in Section V-A, class 'A' is different from other classes, which refers to emotions.

We further divide all the six classes for data1 into two major classes, one is for emotion class and the other is for non-emotion class. Abbreviation *TC* is used to refer to the IC task that involves two classes. *A* refers to a emotion

[3] "一七六九八"(one seven six nine eight) to 17698
[4] http://ictclas.nlpir.org/newsdownloads?DocId=352
[5] http://www.sogou.com/labs/dl/c.html

TABLE 2. Experiments results for individual classifiers and ensembles . ∗ indicates this result is significantly better than the result before it in this column with $p < 0.001$ of the T-test.

	Data1				Data2			
	EA Acc	EA Re	TC Acc	Final Acc	EA Acc	EA Re	TC Acc	Final Acc
CNN	89.31	89.78	90.22	76.25±0.27	83.67	83.15	79.35	66.54±0.15
LSTM	90.38	89.77	90.56	78.03±0.51	84.97	85.62	81.89	69.59±0.62
DCNN	94.59	94.53	95.75	80.73±0.29	85.32	88.67	92.26	72.49±0.25
HAN	94.27	94.95	95.53	82.61±0.42	86.52	91.06	**94.60**	77.82±0.38
HAN-R	-	-	-	-	88.88	87.83	92.09	78.66±0.19
CNN-LSTM	92.46	92.22	93.01	79.87±0.26	87.15	75.61	83.26	70.23±0.27
MCNN-BLSTM	**96.19**	94.57	94.32	81.19±0.64	88.62	68.23	80.16	73.54±0.58
MCNN-BLSTM-R	-	-	-	-	**91.27**	78.40	86.09	74.93±0.42
BLSTM-2DCNN	94.21	94.56	92.34	81.08±0.53	90.38	78.46	81.66	75.64±0.61
RF	94.03	94.81	95.39	83.77±0.40	90.13	90.55	85.21	78.10±0.43
SRE	93.23	93.08	93.00	81.09±0.51	87.74	88.39	92.33	76.36±0.65
SFE-WITH-CNN-LSTM	93.67	96.02	95.11	83.49±0.37	87.51	89.83	90.56	78.23±0.54
SFE	95.76	**95.98**	**96.96**	84.74±0.21*	87.96	90.61	86.44	79.53±0.51*
SFE-R	-	-	-	-	89.35	**92.01**	87.20	80.75±0.35*
MN-Ada-WITH-CNN-LSTM	94.03	95.98	95.39	83.72±0.31	88.16	87.92	87.16	80.01±0.33
MN-Ada	96.12	95.28	93.00	**85.71**±0.27*	88.44	88.71	86.05	80.86±0.16*
MN-Ada-R	-	-	-	-	90.80	91.51	89.71	**81.83**±0.43*

class, while *EA* refers to a non-emotion class. $Recall_{EA} = oc/(oc + aoc)$, where oc is the number of instances that are correctly classified into *EA*, while aoc is the number of instances that belong to *EA* but are classified into *A*. $Accuracy_{EA} = oc/(oc + ooc)$ where ooc is the number of instances that belong to *A*, but are classified to *EA*. The *TC Acc* and *Final Acc* respectively reflects the accuracy of classifiers for two classes and all classes. As we can see, the MCNN-BLSTM and MCNN-BLSTM-R achieve best results on EA accuracy in data1 and data2, respectively. It shows MCNN-BLSTM can capture sentiment information in dialogues well. In Table 2, we also show the comparison between MCNN-BLSTM and CNN-LSTM. It is easy to see that MCNN-BLSTM is more suitable for our IC task, which can further improve the final accuracy respectively by 1.31% and 3.31% on data1 and data2. HAN performs well on all metric and get best result on TC accuracy in data2, and get best final accuracy in data1 compared with all single model. It shows the effectiveness of hierarchical structure with attention on capturing information. Although MCNN and DCNN do not perform prominent, the gaps from other single models are not significant on every metrics.

In data2, when comparing MCNN-BLSTM, HAN with MCNN-BLSTM-R, HAN-R, MCNN-BLSTM-R, HAN-R get better results on final accuracy. In particular, there is 4.39% increase in the final accuracy for MCNN-BLSTM-R and 0.84% for HAN-R, which get the best result in single model performance in data2. MN-Ada-R, which ensemble individual models with "-R" has a 0.97% increase on the basis of MN-Ada. SFE-R has a 1.22% increase on the basis of SFE. These results show role information is important in IC task and the function we add role information is effective. According to the statistical significance test, we can see the ensemble models we propose all significantly better than other ensemble functions and all single models on final accuracy. The MN-Ada is also significantly better than SFE on final accuracy.

We also ensemble two of four individual classifiers or three of four in use SFE and MN-Ada. The detailed results shown in Table 3. It is easy to see that the performances of combine arbitrarily two classifiers, which show in Table 3 are all better than individual classifiers in Table 2. The results show in 3 are all better than the Table 3 but lower than combining all individual classifiers in Table 2. As we can see the SFE scheme on four individual classifiers outperforms on three individual classifiers in data1 and data2 by 0.5% and 0.34% on average, respectively. This advantage shown in MN-Ada scheme on data1 and data2 are 0.54% and 0.61%, respectively. From these results, we verify that each single classifier we use makes contribution for this task. We also ensemble other three models with CNN-LSTM, and the result show our MCNN-BLSTM more suitable for this IC task.

Almost all ensemble schemes get higher final accuracy compared with single model except SRE. Compared with the best individual model on data1 and data2, SRE has a decrease of 1.52% and 2.3% in the final accuracy. This result indicates that only use classification results as input to second layer for stacking scheme dose not combine the advantages of each model. The ensemble scheme MN-Ada gets best results on final accuracy, which is 3.1% higher than the best single model and 1.94% higher than the RF ensemble scheme in data1. In data2, MN-Ada-R do not get best results on

TABLE 3. The final accuracy of combine different single model with SFE and MN-Ada. (a) Combine two different single models. (b) Combine there different single models.

(a)

		HAN+ DCNN	HAN+ MCNN-BLSTM	HAN+ BLSTM-2DCNN	DCNN+ MCNN-BLSTM	DCNN+ BLSTM-2DCNN	BLSTM-2DCNN+ MCNN-BLSTM	Average
Data1	SFE	83.54	83.65	83.56	83.21	83.44	83.45	83.475
	MN-Ada	84.57	84.91	84.58	84.16	84.42	84.06	84.45
Data2	SFE	78.91	78.97	79.02	78.87	78.79	78.77	78.89
	MN-Ada	80.02	80.21	80.23	79.85	79.92	79.81	80.01

(b)

		HAN+ DCNN+ MCNN-BLSTM	HAN+ MCNN-BLSTM+ BLSTM-2DCNN	HAN+ BLSTM-2DCNN	DCNN+ MCNN-BLSTM+ BLSTM-2DCNN	Average
Data1	SFE	84.21	84.35	84.31	84.13	84.24
	MN-Ada	85.17	85.19	85.24	84.97	85.14
Data2	SFE	79.19	79.21	79.16	79.11	79.17
	MN-Ada	81.07	81.12	81.32	81.03	81.14

TABLE 4. Model proportion of RF and MN-Ada. The first four columns are the proportions of each model in the RF, and the last six columns are the values in the formula 6 obtained by each model in MN-Ada. The C-L represent model MCNN-BLSTM, and L-C represent model BLSTM-2DCNN.

	α	β	γ	δ	μ^{C-L}	μ^{C-L-R}	μ^{L-C}	μ^{DCNN}	μ^{HAN}	μ^{HAN-R}
Data1	0.35	0.35	0.15	0.15	1.135	*	1.262	1.228	1.321	*
Data2	0.8	0.13	0.05	0.12	0.900	0.855	0.924	0.918	1.106	1.142

other metrics, but get the best final accuracy, which is most important in IC task. Although SFE did not achieve the best final accuracy in both datasets, it also get second only to MN-Ada on final accuracy.

In Table 4 we listed the weights of each model in RF and the μ value in MN-Ada of each model. These values are the best experimental results corresponding to. The larger the values the more import the model is in ensemble schemes. This conclusion is correspond to the result of individual classifier.

VI. CONCLUSION

In this paper we propose a new task IC, which is to classify the full dialogue into a category according to the real intention of a customer. It is different with DA task which is classify each sentence to a category to assist in generating smooth dialogue. DA task is depending on the change of entities when IC task is depending on semantics information. Based on these differences, the solutions for DA task often need to combine simple classifier and conditional random field (CRF) to obtain entities, while the solutions of IC task need to apply complex neural network to obtain exact semantics information. We use Chinese dialogue datasets to conduct intention classification. Based on the characteristics of datasets, we propose two ensemble functions SFE and MN-Ada, they both get competitive performance on all metrics especially the final accuracy on both datasets. We also adapt four DC models to our ensemble function. Especially we modify CNN-LSTM to MCNN-BLSTM, which has multi-channel CNN and bidirectional LSTM. The experiment result demonstrate that our improvement is effective. We add dialogue role feature to HAN and MCNN-BLSTM, and get a certain progress, compared with role feature removed. When we ensemble features from each individual classifier, which add role feature, we get state-of-the-art final accuracy on MN-Ada.

APPENDIX A
DIALOGUE EXAMPLE

A. HANDLE

A:很高兴为您服务。
(Glad to serve you.)
B:您好, 能帮我查一下这个月手机消费了多少钱嘛?
(Hello, can you help me to find out how much money is spent on mobile phones this month?)
A:好的, 您这个月消费58.1元。
(Ok, you spend 58.1 yuan this month.)
B:好的, 那我想换个套餐, 因为这张卡很少使用。
(Ok, then I want to change the package because this card is rarely used.)
A:可以呀, 有一款十八元的套餐, 包括一百分钟的通话时长和一百兆流量。

(Yes, there is a 18 yuan package that includes a one hundred minutes of talk time and one hundred megabytes of traffic.)
B:那请帮我换一下吧。
(Then please help me change it.)
A:您是确定要换这个套餐吗?
(Are you sure you want to change this package?)
B:对
(Yes.)
A:好的已经帮您更改成功了。
(OK, I have helped you change successfully.)
B:谢谢。(Thank you)

B. INQUIRE

A:您好, 请讲。
(Hello, glad to serve you.)
B:您好, 我想把这个八十元的套餐改小一点可不可以?
(Hello, I want to change the 80 yuan package to a smaller one.)
A:您请稍等, 我帮您看一下。 您如果需要更改更低的套餐可以办理下个月生效。
(Please hole on. If you need to change the smaller package, you can apply for the next month.)
B:那请问最低是多少呢?
(So what is the minimum package for spending?)
A:是85元的, 包括一百分钟通话和五百兆流量。
(It is 58 yuan, including one hundred minutes of calls and five hundred megabytes of traffic.)
B:那请帮我办五十八的那个吧。
(I want this one)
A:好的, 请稍等。
(OK, please wait a moment)
A:不好意思先生, 查询到您当前有一个赠送的活动没有结束, 无法变更套餐呢。
(Sorry sir. I have found that you have a gift package that is not finished yet. You cannot change the package.)
B:这样啊, 那怎么办啊。
(So, what should I do?)
A:您可以在活动结束后变更套餐。
(You can change the package after the gift package.)
B:好的, 谢谢。(Ok, Thank you.)

C. COMPLAINTS

A:您好, 我是正在断很高兴为您服。
(Hello, glad to serve you)
B:您好, 我想问一下我的话费使用情况。
(Hello, I want to ask about my phone bill usage)
A:好的, 请稍等, 正在为您查询。
(Ok, please wait)
A:您的话费余额不足。
(Your phone bill balance is insufficient)
B:你说啥, 我听不见, 信号不好。
(What are you talking about? I can not hear you. The signal is not good.)
A:您在什么位置?
(Where are you?)
B:是这样的, 现在我就给我们现在有那个杨高北路中泉路与蒙城关镇这里, 然后都没打几个信号好。
(Ok, now I am at Yang Gao north road Zhong Quan road. There is no signal in this place.)
A:知道了。对。杨高北路是吧。
(Ok,I know. YangGao north road right?)
A:您请稍等一下, 我帮您查讯一下。
(Please hold on, i'll check you for help.)
B:好的。
(Ok)
A:好先生是这样子, 非常感谢您对移动公司关注, 我们会第一时间改善当地的信号问题。
(Ok, sir. Thank you very much for your attention to the China Mobile company. We will improve the local signal problem as soon as possible.)
B:好是确实是你们这里有问题了。
(There are some problems in your system indeed.)
B:这个问题我已经反应了好几次了, 你们都没解决。
(I have already responded to this question several times and you have not solved it.)
A:非常抱歉, 我们会尽快解决。
(I am very sorry, we will solve it as soon as possible.)
B:每次都这么说。
(Say so every time.)

APPENDIX B
CHANGE OF DIALOGUE INTENTION

We list the change of dialogue intention for the three different examples we show in Appendix A in Table 5. From the Table 5 we can see our model can capture the change of intention.

TABLE 5. This table show the result of each example when we select different number of sentences. The first line is the sentence numbers we select in a dialogue. The I, H and C represent Inquire, Handle and Complain, respectively.

	2	4	6	8	10	12	14
A	I	I	H	H	H	H	
B	I	I	H	I	I	I	
C	I	I	I	I	I	C	C

Acknowledgment

This work is partially supported by the National Natural Science Foundation of China (Nos. 11590770-4, 61650202, 11722437, U1536117, 61671442, 11674352, 11504406, 61601453), the National Key Research and Development Program (Nos. 2016YFB0801203, 2016YFC0800503, 2017YFB1002803) and the Key Science and Technology Project of the Xinjiang Uygur Autonomous Region (No. 2016A03007-1) and Foundation of Science and technology on Information Assurance Laboratory(No. KJ-17-102).

REFERENCES

[1] S. Wang and C. D. Manning, "Baselines and bigrams: Simple, good sentiment and topic classification," in *Proc. Meeting Assoc. Comput. Linguistics*, 2012, pp. 90–94.
[2] L. Zhang and B. Liu, "Sentiment analysis and opinion mining," *Synthesis Lect. Hum. Lang. Technol.*, vol. 30, no. 1, pp. 152–153, 2016.

[3] R. Kant, S. H. Sengamedu, and K. S. Kumar, "Comment spam detection by sequence mining," in *Proc. 5th ACM Int. Conf. Web Search Data Mining*, 2012, pp. 183–192.
[4] L. M. Manevitz and M. Yousef, "One-class SVMs for document classification," *J. Mach. Learn. Res.*, vol. 2, pp. 139–154, Dec. 2001.
[5] N. Kalchbrenner, E. Grefenstette, and P. Blunsom. (2014). "A convolutional neural network for modelling sentences." [Online]. Available: https://arxiv.org/abs/1404.2188
[6] P. Liu, X. Qiu, X. Chen, S. Wu, and X. Huang, "Multi-timescale long short-term memory neural network for modelling sentences and documents," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2015, pp. 2326–2335.
[7] G. Wang, J. Sun, J. Ma, K. Xu, and J. Gu, "Sentiment classification: The contribution of ensemble learning," *Decis. Support Syst.*, vol. 57, pp. 77–93, Jan. 2014.
[8] A. Onan, S. Koruko lu, and H. Bulut, "Ensemble of keyword extraction methods and classifiers in text classification," *Expert Syst. Appl.*, vol. 57, pp. 232–247, Sep. 2016.
[9] T. G. Dietterich, "Ensemble methods in machine learning," in *Multiple Classifier Systems*. Berlin, Germany: Springer, 2000.
[10] L. Brieman, "Bagging predictors," *Mach. Learn.*, vol. 24, no. 2, pp. 123–140, 1996.
[11] Q. Wu, Y. Ye, H. Zhang, M. K. Ng, and S.-S. Ho, "ForesTexter: An efficient random forest algorithm for imbalanced text categorization," *Knowl.-Based Syst.*, vol. 67, no. 3, pp. 105–116, 2014.
[12] P. Bühlmann and B. Yu, "Analyzing bagging," *Ann. Statist.*, vol. 30, no. 4, pp. 927–961, 2002.
[13] Y. Freund and R. E. Schapire, "A decision-theoretic generalization of online learning and an application to boosting," *J. Comput. Syst. Sci.*, vol. 55, no. 1, pp. 23–37, 1997.
[14] L. Breiman, "Stacked regression," *Mach. Learn.*, vol. 24, no. 1, pp. 49–64, 1996.
[15] H. Drucker, C. Cortes, L. D. Jackel, Y. Lecun, and V. Vapnik, "Boosting and other ensemble methods," *Neural Comput.*, vol. 6, no. 6, pp. 1289–1301, 1994.
[16] D. Opitz and R. Maclin, "Popular ensemble methods: An empirical study," *J. Artif. Intell. Res.*, vol. 11, pp. 169–198, Aug. 1999.
[17] L. Breiman, "Random forests, machine learning 45," *J. Clin. Microbiol.*, vol. 2, pp. 199–228, Jan. 2001.
[18] J. Zhu, H. Zou, S. Rosset, and T. Hastie, "Multi-class AdaBoost," *Statist. Interface*, vol. 2, no. 3, pp. 349–360, 2009.
[19] C. Ju, A. Bibaut, and M. V. D. Laan, "The relative performance of ensemble methods with deep convolutional neural networks for image classification," *J. Appl. Statist.*, vol. 45, no. 15, pp. 2800–2818, 2017.
[20] T. Strauss, M. Hanselmann, A. Junginger, and H. Ulmer. (2017). "Ensemble methods as a defense to adversarial perturbations against deep neural networks." [Online]. Available: https://arxiv.org/abs/1709.03423
[21] Y. Kim. (2014). "Convolutional neural networks for sentence classification." [Online]. Available: https://arxiv.org/abs/1408.5882
[22] W. Yin and H. Schütze, "Multichannel variable-size convolution for sentence classification," in *Proc. 19th Conf. Comput. Lang. Learn.*, 2016, pp. 204–214.
[23] J. Y. Lee and F. Dernoncourt, "Sequential short-text classification with recurrent and convolutional neural networks," in *Proc. NAACL-HLT*, 2016, pp. 515–520.
[24] C. Zhou, C. Sun, Z. Liu, and F. C. M. Lau, "A C-LSTM neural network for text classification," *Comput. Sci.*, vol. 1, no. 4, pp. 39–44, 2015.
[25] J. Wang, L. C. Yu, K. R. Lai, and X. Zhang, "Dimensional sentiment analysis using a regional CNN-LSTM model," in *Proc. Meeting Assoc. Comput. Linguistics*, 2016, pp. 225–230.
[26] Z. C. Lipton, D. C. Kale, C. Elkan, and R. Wetzel. (2015). "Learning to diagnose with LSTM recurrent neural networks." [Online]. Available: https://arxiv.org/abs/1511.03677
[27] 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.*, 2017, pp. 1480–1489.
[28] A. Conneau, H. Schwenk, L. Barrault, and Y. Le Cun, "Very deep convolutional networks for text classification," in *Proc. 15th Conf. Eur. Chapter Assoc. Comput. Linguistics*, 2016, pp. 1107–1116.
[29] P. Zhou, Z. Qi, S. Zheng, J. Xu, H. Bao, and B. Xu. (2016). "Text classification improved by integrating bidirectional LSTM with two-dimensional max pooling." [Online]. Available: https://arxiv.org/abs/1611.06639

MANSHU TU was born in Handan, Hebei, China, in 1991. She received the B.S. degree in biomedical engineering from Beijing Jiaotong University, China, in 2014. She is currently pursuing the Ph.D. degree with the Key Laboratory of Speech Acoustics and Content Understanding, Institute of Acoustics, University of Chinese Academy of Science. Her current research interests include natural language processing, transfer learning, and sentiment analysis.

BING WANG received the Ph.D. degree in computer science from the University of New South Wales, Australia, in 2014. She is currently an Associate Professor with the Institute of Acoustics, Chinese Academy of Sciences, China. Her research interests include data mining, machine learning, and social computing.

XUEMIN ZHAO received the B.S. degree in communication engineering from Nankai University, China, in 2006, and the M.S. and Ph.D. degrees from the Key Laboratory of Speech Acoustics and Content Understanding, Institute of Acoustics, University of Chinese Academy of Science, where he has been an Associate Researcher, since 2014. His research interests include natural language processing and digital audio watermarking.