Call For Help Detection In Emergent Situations Using Keyword Spotting And Paralinguistic Analysis

Huangrui Chu*
Yechen Wang*
huangrui.chu@dukekunshan.edu.cn
yechen.wang@dukekunshan.edu.cn
Data Science Research Center, Duke
Kunshan University
Kunshan, Jiangsu, China

Ran Ju
Data Science Research Center, Duke
Kunshan University
Kunshan, Jiangsu, China

Yan Jia
Data Science Research Center, Duke
Kunshan University
Kunshan, Jiangsu, China

Haoxu Wang
Data Science Research Center, Duke
Kunshan University
Kunshan, Jiangsu, China

Ming Li†
Data Science Research Center, Duke
Kunshan University
Kunshan, Jiangsu, China

Qi Deng
qi.deng@kone.com
Technology Asia and Escalator, KONE
Elevators Co., Ltd.
Kunshan, Jiangsu, China

ABSTRACT

Nowadays, the safety of passengers within the enclosed public space, such as the elevator, becomes more and more important. Though the passengers can click the "SOS" button to call the remote safety guard, the chances are that some passengers might lose their ability to stand up to click the button or it is not convenient to do so when in emergency situations. Also, people's first reaction may be to call for help using voice instead of pressing the mayday button. Thus, we believe a speech-based system is very useful under this scenario. This work proposes a system using keyword spotting and paralinguistic analysis to detect whether the passenger calls for help in mandarin and gives real-time feedback which might provide the passenger within time help to prevent the accident. Unlike the standard keyword spotting task which is to detect the predefined call for help keyword "jiu ming" in any scenario, we focus on detecting both the keyword and the paralinguistic states. The system will only be triggered when the keyword and the emergency situation such as shouting or screaming appear at the same time. To this end, we compared the performance of different methods and we find that the deep neural network-based small-footprint keyword spotting methods are effective and efficient for keyword spotting tasks under emotional scenarios.

CCS CONCEPTS

• **Human-centered computing**;

KEYWORDS

Keyword Spotting; Paralinguistic Analysis; Deep learning; Machine Learning

1 INTRODUCTION

Keyword spotting (KWS), a task to detect whether a specific word or phrase appears in a continuous speech, is widely applied in our daily life. Applications of wake-up modules on smart phone assistants and smart speakers like "Hi, Siri" from Apple and "Ok, Google" from Google achieve satisfying performance with an energy-efficient implementation. Paralinguistic Analysis (PA) is used to study the paralinguitic speech attributes,e.g. emotion, age, gender, mental states and much other information that humans conveyed in communication. Recently, there has been a growing demand for algorithms that can recognize a particular keyword and generate adjusted solutions according to the real scenario. For example, to secure the safety of humans in an enclosed space such as the elevator, we may need to develop a system that can efficiently detect whether the person is calling for help in a real emergent situation. However, it is not robust enough to identify an emergency only by identifying specific keywords as much of the content spoken in the emergency could also show up in daily life communication. Comparing real emergency speech and normal verbal communication, we believe that the former contains more emotional paralinguistic characteristics. Therefore, we want to include both keyword and paralinguistic information into our consideration. To this end, we want to measure how the paralinguistic information would affect the performance of the different KWS systems and which technique is more suitable in developing tasks that needs to consider paralinguistic information.

The traditional KWS systems are made up of the large vocabulary continuous speech recognition (LVCSR) module followed by a keyword spotting module that can look up whether the keyword appears in the lattice generated by the LVCSR module [Chen et al. 2013]. The benefit of this procedure is that the accuracy is relatively high and the workload of changing the customized keyword is relatively low since we can reuse the LVCSR module. Although, after specific training, this method could recognize the content of the conversation expressed with an extreme emotion like painfulness and scarceness, it requirs large scale emotional training data in real application. It thus results in significant cost and inefficiency in the real application. Therefore, this method is unsuitable for some applications, especially when deployed in environments with low computing resources and target for real time application.

More recently, the DNN based small footprint approaches have become more popular [Chen et al. 2014]. Those approaches are

*Both authors contributed equally to this research.
†Corresponding Author.

usually made in 3 steps. Firstly, the speech features such as Mel-Frequency Cepstrum Coefficients (MFCC)) [Yang et al. 2014] and log-Mel filter bank energy (Fbank) are extracted. A deep neural network (DNN) [Chen et al. 2014], and its variants will then be trained to predict the word-level or phoneme-level probabilities of appearance. Neural network architectures such as convolutional neural network (CNN) [Sainath and Parada 2015], long-short term memory (LSTM) [Sun et al. 2016], and attention mechanisms [Shan et al. 2018] are also explored and they show better result. Finally, the posterior probability generated by the DNN model will be used to calculate the confidence score. Those DNN based small footprint approaches are more widely used in our real-life applications such as the wake up module used in smart assistants in smart phones and smart speakers since they usually cost less computational resources and have the advantage of low latency. As a result, we choose the DNN based approach to build our KWS module.

Paralinguistics, which means 'alongside linguistics', is confined to the realm of human to human communication, but with a broad and a narrow meaning[Schuller et al. 2013]. The narrow meaning, as Crystal says, restricts the scope of the term to "vocal factors involved in paralanguage" [Crystal 2019]. At the same time, the broad meaning also involves facial expression as an element of paralanguage. Nevertheless, the word "vocal factor" itself is not well-defined, with a narrowing meaning excluding linguistic/verbal factors or a broad meaning including them [Schuller et al. 2013]. In this paper, we define paralinguistic analysis as dealing with the phenomena modulated onto or embedded into the verbal message.

Researchers come up with many features to extract such information to study the vocal cues conveyed in the speech. For example, MFCC and Cepstrum [Lalitha et al. 2015] are widely used in the study of the emotion of the speaker. Features like F0, F1, F2, F3, H1-H2, H2-H4, H4-H2K inspired by the psychoacoustic model of voice quality [Park et al. 2017] and i-vectors [Dehak et al. 2011a] are used in speaker verification tasks. The features used in the paralinguistic analysis can be roughly classified into two classes: long-term (mostly prosodic) features and short-term features based on MFCCs. Considering that the main task for us is to determine whether the passenger encounter emergency in that specific period, short-term features like MFCC are preferred. Since there is no need to differentiate the speakers' identity and the task should only focus on whether the passenger just casually mentions the keyword "jiu ming" or desperately asks for help, MFCC is adopted here.

Support Vector Machines (SVM) with kernel functions that use a task-coupling parameter[Evgeniou and Pontil 2004], Gaussian mixture model (GMM) [Vondra and Vích 2008], and DNN [Snyder et al. 2017] are explored when doing paralinguistic analysis. Recently, CNN is also applied in paralinguistic analysis [Chlasta et al. 2019]. Since residual neural network (ResNet) [He et al. 2016] is widely used as base classifiers in the current classification task, we choose ResNet as the backbone for our paralinguistic analysis.

Since we need to take the consideration of both keyword information and paralinguistic information, we assume a two-stage approach that combines both a KWS module and a PA module will have better performance. In the first stage, the LSTM based KWS module will judge whether the keyword "jiu ming" appears. The second stage uses an end-to-end classifier with the residual neural network(Resnet)[He et al. 2016] as the PA module backbone structure. A valid call for help event must pass through both stages. We compare this system with the other implementations, including a Kaldi [Povey et al. 2011] based LVCSR KWS approach, a KWS only binary classification system and a KWS only multi-classification system, which only uses keyword spotting to detect the real call for help scenario.

This paper is organized as follows. The task definition, the datasets, detailed description of our system, and implementation details are shown in Materials and Methods. The evaluation metrics and experimental results are shown in Results. The interpretation of the result and the findings and implication is provided in Discussion.

2 MATERIALS AND METHODS

2.1 Task Definition and Proposed Dataset

2.1.1 Task Definition. Our goal is to build a system that can automatically detect the keyword "jiu ming" and recognize the paralinguistic patterns in an emergent of calls for help situations in the elevator scenes thus improve the efficiency of the safety monitoring system. This task is targeting online applications and therefore needs to be energy-efficient and easy to deploy. However, considering some common conversations might also include the keyword while the passengers just have a casual talk or mimic a scene from the films, there is a need to differentiate the real or fake call for help within this scene. The system is supposed to be triggered when the real call for help under emergent situations is detected. Therefore, we consider the daily life scene and compose scripts to define the real and fake call for help category. The call for help is categorized as real if passengers speak the keyword "jiu ming" and they need rescues in some urgent emergencies. For instance, the passenger is tracked in the elevator or robbed by other people. A fake call for help means the keyword "jiu ming" is detected in the conversation, while passengers do not request rescues. For instance, a man would say:" I heard that someone shouted 'jiu ming!' in the car accident. Did you hear about that news?" In this example, though the man might imitate the person's call for help, he does not have the motivation of asking for rescues because he calmly stays in the elevator. On account of confusing cases like that, besides the auto-detection task, there is a need to propose such a real or fake call for help recognition task that can deal with this kind of practical need.

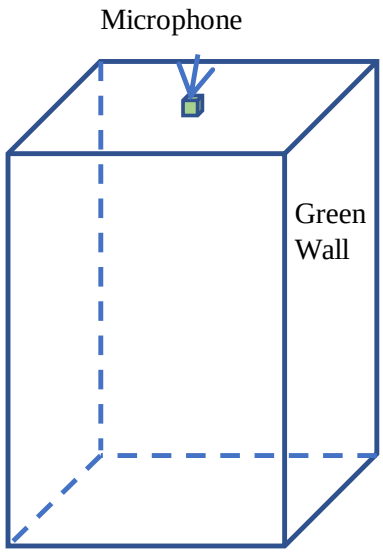

Figure 1: The scenes we collect the dataset with one microphone in the center of the ceiling.

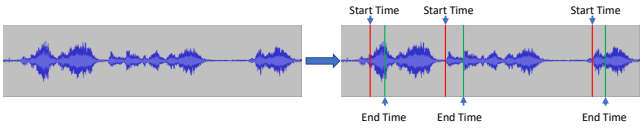

Figure 2: We annotate the start point and end point of the keyword "jiu ming".

Table 1: Selective Scripts for performing acted call for help events

Contents of the conversation(Speaking in Mandarin)
"Jiu ming! Help me."
"The elevator suddenly stopped! Jiu ming!
"I'm trapped. Is there anyone outside? Jiu ming!"
"Jiu ming! Where is the security? Please help me."
"Someone lost consciousness! Jiu ming!"
"Jiu ming! Come and save him. Call 120!"
"I have a heart attack. Jiu ming!"
"I'm dying. Jiu ming!"
...

Table 2: Details of Datasets for the whole task

Dataset	Num of session	Num of clips
Training set	186	47051
Validation set	27	6433
Test set	53	24455

2.1.2 The Proposed Dataset. The proposed dataset is initially collected by us using the method described below. As shown in Figure 1, the audio part of our acted dataset is collected by a low price microphone with a sampling rate of 16 kHz in a simulated elevator environment.

Two hundred sixty-six actors were employed and provided with selective scripts for performing various scenarios (shown in Table 1). In each real call for help event, the actor(s) could randomly choose a dialogue and freely play according to their understanding of the events. For the fake call for help, actors could talk about some topics with "jiu ming" without any restriction. For example, the actor may say, "Did you hear the new album released yesterday? The name of that album is 'jiu ming'." to another one. The entire dataset is completed based on a combination of given scripts for reference and freestyle performance. To improve the data quality, we manually annotate the start point and end point of each keyword "jiu ming" in all the audio data, as shown in Figure 2. Besides the call for help data, we also collect data in some daily scenes in the elevator that does not contain the keyword "jiu ming". These data were used to train the keyword spotting system where the keyword spotting system is not supposed to wake up when getting the audio data from this category. We split the whole dataset into three subsets: training set, validation set, and test set at a ratio of 7:1:2 with respect to different sessions as shown in Table 2, where each session having a main character different from the other.

We pre-process the data in the training set for different systems. For the KWS module, the keyword data is cut according to the start point and the end point of "jiu ming". For the PA module, the input audio data is cut to a 1-second clip (centered at the keyword "jiu ming"). We call each cut audio segment a "clip". We first find the start point and end point of the keyword, adding 0.5 seconds at the left and right of the time stamp, and extract the keyword speech as the new positive sample clips. Since the average length of the keyword is around 445 ms, we will eventually have keyword samples each of around 1.5 seconds. Whether the keyword is real or fake is also annotated for different tasks. For the negative samples, we also divide them into around 1.5 seconds of speech. The detail of how many clips are there in different systems is shown in Table 3, Table 4, Table 5, and Table 6, respectively. We also show their label and which part of the whole initial dataset they are from.

2.2 System Description

We investigate different approaches in our experiment, including the Kaldi [Povey et al. 2011] based LVCSR+KWS approach, the LSTM based small-footprint KWS + PA approach, the LSTM based small-footprint KWS only binary classification approach, and the LSTM based small-footprint KWS only multi-classification approach.

2.2.1 The Kaldi LVCSR-KWS Approach. We used the traditional LVCSR-KWS approach as the baseline system. The whole system was developed using Kaldi [Povey et al. 2011]. For the LVCSR model, we use a large amount of data to train the chain model with the factorized Time delay neural network (TDNNF) [Povey et al. 2018] and its generated lattice will be decoded using the Kaldi online decoding style to determine if the keyword exists.

Table 3: Detailed information about data used in KWS module training within the KWS+PA framework

Dataset	Label	Num of clips	From which part of the whole initial dataset
Train	Postive	8188(5790+2398)	Train set real & fake keyword "jiu ming" speech
	Negative	38863	Train set nonkeyword speech
Valid	Postive	1265(842+423)	Valid set real & fake keyword "jiu ming" speech
	Negative	5168	Valid set nonkeyword speech

Table 4: Detailed information about data used in PA module training within the KWS+PA framework

Dataset	Label	Num of clips	From which part of the whole initial dataset
Train	Postive	5379	Train set real keyword "jiu ming" speech
	Negative	2305	Train set fake keyword "jiu ming" speech
Valid	Postive	753	Valid set real keyword "jiu ming" speech
	Negative	409	Valid set fake keyword "jiu ming" speech

Table 5: Detailed information about data used in KWS module training within the KWS Only Binary Classification framework

Dataset	Label	Num of clips	From which part of the whole initial dataset
Train	Postive	5790	Train set real keyword "jiu ming" speech
	Negative	41261(2398+38863)	Train set fake keyword & nonkeyword speech

Table 6: Detailed information about data used in KWS module training within the KWS Only Multiclassification framework

Dataset	Label	Num of clips	From which part of the whole initial dataset
Train	Real	5790	Train set real keyword "jiu ming" speech
	Fake	2398	Train set fake keyword "jiu ming" speech
	Others	38863	Train set nonkeyword speech

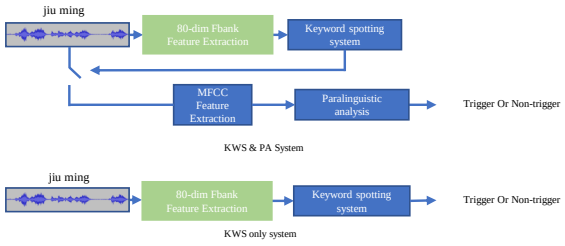

Figure 3: The overview frameworks of the proposed systems.

2.2.2 The LSTM-KWS module. For the KWS module, we choose an LSTM based approach with small footprint and low latency. The predefined keyword "jiu ming" only contains two Chinese characters, and as a result, it may only appear in the speech for a very short time. Thus, we choose to implement a binary classification method whose modeling unit contains a filler unit representing the non-keyword speech and the keyword unit itself. The keywords are cut out from the initial audio recording as the positive samples in our training data. We then calculated the average length of the keyword speech as the input length of the network. We extract 80-dimension Fbank features as the input of the following neural network. We adopt a two-layer stacked LSTM followed by an average pooling layer and a final fully-connected layer as the main component of our neural network. The neural network has the output dimension of 2, which stands for the non-keyword speech and keyword speech probability. The KWS module gets triggered if the output keyword speech probability passes the pre-defined threshold.

2.2.3 The PA module. In this part, we treat the proposed task as an audio-based classification problem since the keyword has already been found in the LSTM-KWS module and we extracted the keyword as the input of the PA module. The MFCC of the extracted data is computed as the pre-processed feature and then undergoes the standardization process. We adopt the ResNet-18 [He et al. 2016] as the backbone network for processing the standardized MFCC feature and return the confidence values of the input audio data as a real/fake call for help, respectively. By comparing the confidence values of these two categories, the PA module will decide whether the passenger is calling for help in a real emergency.

2.2.4 The LSTM-KWS+PA system. We combine the KWS module and a PA module to develop the LSTM based small-footprint KWS + PA approach as our proposed system. The overall procedure of our system is shown in Figure 3. We first extract the 80-dimension Fbank speech feature and feed it into our KWS module for the given speech. If the KWS module is triggered, the PA module will be activated. We find the end time of the keyword speech given by the KWS module and use its previous 1 second speech to extract MFCC features, which will then be used as the input of the PA model. If it successfully passes both the KWS module and the PA module, we will consider that at this time point, the speech contains the emotional call for help speech.

2.2.5 Keyword Spotting Only Binary Classification System. We add the small-footprint binary classification approach as a comparison to test its performance under this specific task and quantify the advantage and disadvantages of the LSTM-KWS+PA system. Similar to the LSTM-KWS module, we use the 80-dimension Fbank as the input feature and two-layer stacked LSTM whose output indicates the posterior probability of the real keyword and the others.

2.2.6 Keyword Spotting Only Multi-classification System. For the same reason, we add the small-footprint multi-classification approach as a comparison to test its performance under this specific task for further comparison and measurement. It also follows a similar procedure to the previous LSTM based systems. The difference is that the output dimension of the LSTM model is 3, representing the posterior probability of non-keyword speech, fake keyword, and real keyword speech in an emergency.

2.3 Implementation Details

The implementation details for different approaches are described below:

2.3.1 The Kaldi LVCSR-KWS Approach. The LVCSR was trained using multiple Chinese corpus on OpenSLR, including aidatatang_200zh [Beijing DataTang Technology Co. 2006], Aishell [Hui Bu 2017], Magic Data [Magic Data Technology Co. 2019], Primewords, [Primewords Information Technology Co. 2018], ST-CMDS [stc [n. d.]], and THCHS-30 [Wang and Zhang 2015]. Few private datasets are also used to train this model and all the corpus we use after various kinds of data augmentation approaches sum up to around 16 thousand hours of speech. We use the 40-dimensional MFCC as the input features. The 15 layers TDNNF based chain model is applied to model the triphone unit and trained with Kaldi script. The keyword "jiuming" is defined by the following grammar "J_B I_4_I OO_4_I U_4_I M_I I_4_I NG_4_E". As for the lattice indexing, we use the online decoding style which uses the LCVSR model to generate lattice and decode the speech every 0.25 seconds.

2.3.2 Keyword Spotting and Paralinguistic Analysis System. This two-stage system contains a KWS module and a PA module, and each module is trained and optimized separately.

For the KWS module, we manually annotated the timestamp of the starting point and endpoint of all the keyword speech in the training set and cut them out as the positive keyword training samples. The rest of the speech will be spliced together and added to the non-keyword negative training samples. We also add aidatatang_200zh [Beijing DataTang Technology Co. 2006] and aishell [Shi et al. 2020] as negative samples. To make the model more robust and suitable for real life applicationm, we apply data augmentation methods such as adding MUSAN [Snyder et al. 2015] noise, music, babble and reverberation. We calculated the average length of the keyword speech in the training set, which is around 445ms, and use it as our sequence length for feature extraction. We choose the 80-dimension Fbank features with a 25ms window length and a shift of 10ms as the input feature of our model. With a 445ms window as the input sequence length, we can know that the input dimension of our neural network is 42*80. For the positive samples, silence will be added to both sides of the signal if it is too short. If it is longer than 445ms, we choose the part in the middle. We cut them into around 1-second pieces for the negative samples and extract the full 1-second features. We will randomly choose a part of the 1-second feature during the training process to form the input feature that represents 445ms. This method acts as a data augmentation method to obtain as many negative samples as possible without overlapping and losing diversity. We construct a two-layer stacked LSTM model with a hidden size of 128. It is followed by an average pooling layer and a fully-connected linear layer. A more complex model could help improve the result, but there is no need to do so in our research. The model was trained for 100 epochs using a Stochastic Gradient Descent method with Nesterov momentum, and the lost function is set to be a cross-entropy loss. We initialize the learning rate at 0.05, and every time the training loss converges, it will decrease by the factor of 0.1. The batch size equals 1024. After the training, we choose a threshold by considering the performance on the valid set. For each given speech, we first extract its 80-dimension Fbank features. Then a sliding window method which has a window length of 42 and step length of 3, is used to produce the LSTM output sequence of 2-dimension posterior probabilities. Finally, we find the maximum probability of keyword appearance from the sequence, and it will be considered as the probability of having the keyword in this given speech. If this probability passes the threshold, we believe it contains the keyword. With that information, we optimized our model by adjusting our threshold until the number of false alarms in each hour is equal to 5 on the validation set.

For the PA module, since the main purpose is to apply this model in real life, data augmentation which could increase the diversity of the existing training data and make it closer to real-life data is applied to enrich our training set and mitigate overfitting. Similar to N. Dehak et al. in [Dehak et al. 2011b], volume augmentation, additive noise, and reverberation, which involves convolving room impulse responses (RIR) with audio, are employed in our data augmentation. We employed the simulated RIRs of the large room, medium room, and small room, which are described by Ko et al. in [Ko et al. 2017]. Regarding additive noise, we use Gaussian noise and MUSAN noise [Snyder et al. 2015], which consists of over 900 noises, 42 hours of music from various genres, and 60 hours of speech from twelve languages.

To augment a recording with MUSAN or RIR, we choose between one of the following randomly:

• reverb: The training recording has artificially reverberated via convolution with simulated RIRs.

• noise: A single audio file which is randomly selected from “babble,”

"music," "noise" of Musan will be trimmed to match 1 second and then added to the original signal.

The input audio is set to last for 1 second with "jiu ming" in the middle of the audio data. We first do the data augmentation as shown above for the 1 second data and then extract the MFCC feature from it using mfcc function imported from python_speech_features with a 25 ms window length and a shift of 10 ms. With a 16000 ms length input, we can know that the input dimension of our neural network is 101*13. After extraction, the MFCC feature is then normalized and transformed to tensor. A tenser MFCC with a dimension of size one inserted at the first position is then used as input to Resnet-18 model for a binary classification task. The model was trained for 40 epochs using an Adaptive Learning Rate method with Adadelta algorithm. We initialize the learning rate at 1, with coefficient used for computing a running average of squared gradients equal to 0.9, term added to the denominator to improve numerical stability equal to 1e-06, and weight decay equal to 0.00001, and the lost function is set to be a cross-entropy loss. The outcomes are confidence values for real and fake call for help respectively. By comparing the confidence values of these two categories and find the maximum, the PA module will decide whether the passenger is calling for help in a real emergency.

2.3.3 Keyword Spotting Only Binary Classification System. It follows the same procedure as the KWS part in the LSTM-KWS + PA system. Except for the training set, only the real emotional keyword speech in emergency situations are labeled as positive samples. All the fake keyword speech spoken in daily life communications and non-keyword speech are labeled as negative samples. The feature extraction, model architecture, training procedure, and evaluation process remain the same. The final decision is made by finding the label of the maximum posterior probability.

2.3.4 Keyword Spotting Only Multi-classification System. Similarly, it also follows a similar procedure to the previous small-footprint KWS systems. The main difference is still the training set. We manually created three labels according to the speech content and scenario. The keywords spoken in the emergency are labeled as real keyword samples. The fake keywords are labeled as "fake" samples. And the non-keyword speech is labeled as "others" samples. We perform multi-classification training and keep all the other settings the same. The final decision is made by finding the label of the maximum posterior probability.

3 RESULTS

We evaluate the performance of our system on the test set.

As shown in Table 7, the component of the final test dataset includes:

(1) The cut segments which do not include the keyword and other dialogues are labelled as others ;
(2) The cut segments which include the keyword but not in emergency are labeled as fake call for help;
(3) The cut segment which includes the keyword in real emergency is labeled as real call for help.

When training the PA module, we want to see the performance of our PA module in the validation dataset. Therefore, we record the "false alarm rate" and "recall rate" of each epoch of our PA module during the 40 epochs' training. In the end, we come up with two criteria to evaluate and select our PA module. First, we want to minimize the false alarm rate and meanwhile keep the recall rate at a reasonable value which is greater than 0.7 and choose the corresponding parameters from that epoch and save this vision as PA 1. Second, we want to maximize the recall rate and meanwhile keep the false alarm rate at a reasonable value which is smaller than 0.3 and choose the corresponding parameters from that epoch and save this vision as PA 2.

In the end, we have five models as following:

(1) LVCSR+KWS Kaldi System (Kaldi)
(2) Keyword Spotting only Binary Classification System (KWS 2): return two confidence values respectively for the two categories: other, real; In this system, other and fake call for help is the same category;
(3) Keyword Spotting only Multi-classification System (KWS 3): return three confidence values respectively for the three categories: other, fake, real;
(4) Keyword Spotting and Paralinguistic Analysis System 1 (KWS 1 + PA 1): First use the keyword spotting to detect whether the keyword appears in the audio or not. If the keyword appears, then cut 1 second of audio and send to PA 1 to see whether this is real call for help or fake call for help.
(5) Keyword Spotting and Paralinguistic Analysis System 2 (KWS 1 + PA 2): First use the keyword spotting to detect whether the keyword appears in the audio or not. If the keyword appears, then cut 1 second of audio and send to PA 2 to see whether this is real call for help or fake call for help.

For Keyword Spotting Only Binary Classification System, the neural network has the output dimension of 2 for each sliding window, which represents the predicted probability of the label "others and fake" and "real". We applied a softmax function on it and the result of the 'real' label will be considered as the posterior probability of the real keyword appearance in this sliding window. Among the list of posterior probabilities generated from the sliding window features, we take the maximum value of each frame and consider it as the probability of the real keyword appearance in this given clip. If the maximum value appears in the "real" class, we think this clip contains the real call for help.

For Keyword Spotting Only Multi-classification System, the neural network have the output dimension of 3 for each sliding window, which represents the predicted probability of the label "others", "fake" and "real". Similarly, a softmax function is applied to generate posterior probability for sliding window. After we get the final probability of each class, we then see the which class the maximum value belongs to. If in the frames, one has the maximum value belongs to "real", we then think this utterance has real call for help.

For Keyword Spotting and Paralinguistic Analysis System, two steps are involved since it has two separated systems. We first define the evaluation method and determine the thresholds on validation dataset. For each given utterance, we apply a sliding window method after the feature extraction. For each extracted features, we choose 42 as our window size to match the input size of the neural network during training. We choose a step size of 3 to generate the list of sliding window features and feed them into our neural network. For each window, we have the probability of

Table 7: Detailed data used in Test

Dataset	Label	Num of clips	From which part of the whole initial dataset
Test	Real	1714	Test set real keyword "jiu ming" speech
	Fake	744	Test set fake keyword "jiu ming" speech
	Others	21997	Test set nonkeyword speech

Table 8: Overall Performance of our models

Model	F1 Score	Recall	False Alarm
Kaldi	0.3261	0.5313	0.01317
KWS 2	0.9186	0.9352	0.007607
KWS 3	0.9063	0.9428	0.01038
KWS 1+PA 1	0.8081	0.7124	0.003826
KWS 1+PA 2	0.8756	0.8705	0.008883

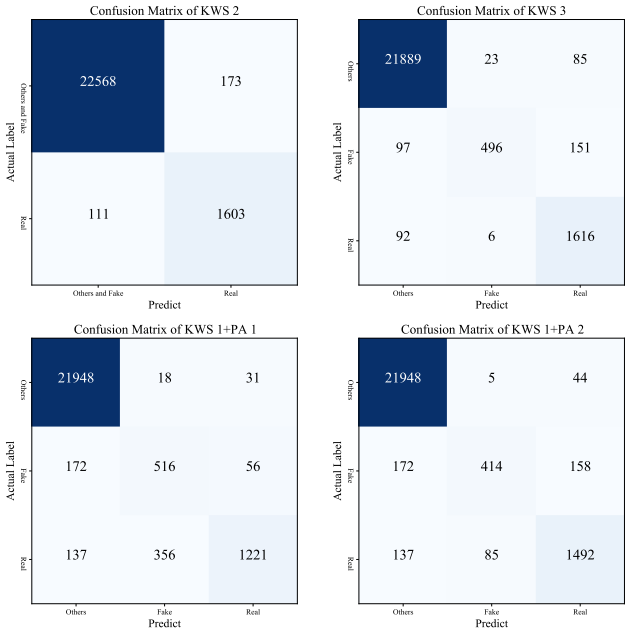

Figure 4: The confusion matrices of our models.

whether the keyword appears in this window. If the probability is larger than the threshold we get using data from valid dataset, then the KWS module is triggered. If the KWS module is not triggered, we will consider the given utterance as non-keyword speech. When the KWS module is triggered, we record timestamp and input the 1 second long audio around the recorded timestamp to the PA system. If the input passes the PA system, we consider this utterance contains real keyword. If not, we believe this utterance contains only fake keyword.

To find a fair method to compare the performance of each model, the F1 score, Recall, False alarm are evaluated with "other and fake" as a class and "real" as a class. The detailed results of all our five models are shown in Table 8. The detailed results of our KWS 1, PA 1, and PA 2 modules are shown in Table 9. The confusion matrices of our models are shown in Figure 4.

$$Recall = \frac{True\ Positive}{True\ Positive + False\ Negative} \tag{1}$$

$$False\ Alarm = \frac{False\ Positive}{False\ Positive + True\ Negative} \tag{2}$$

$$F1\ Score = \frac{2 * True\ Positive}{2 * True\ Positive + False\ Negative + False\ Positive} \tag{3}$$

4 DISCUSSION

From Table 8, we can find that compared with Kaldi, all the LSTM-based KWS systems have a much higher recall rate. It suggests that for the keyword spotting tasks with the emotional scenario, LSTM-based KWS systems would give a better performance. It might be possible that we can fine-tune the LVCSR model using emotional data to get better performance in this task, which means that we will have to adjust the LVCSR model for each specific task. However, it will result in a loss of model reusability, which is the main advantage of the LVCSR KWS approach. Also, a much more training corpus will be needed to obtain an LVCSR model of the same level. Furthermore, in online applications, the small-footprint approaches are usually more energy-efficient and low-latency. Therefore, we believe the small-footprint DNN based approach could be more efficient in training and deployment when dealing with the task that has emotional scenarios in this task.

Considering the difference between four of the small-footprint approaches, we find that the KWS only methods (KWS2 and KWS3) have a slightly better performance. The KWS + PA method could reduce the false alarm rate at the cost of reducing the recall rate. We believe the main reason why the KWS+PA can not outperform the KWS2 and KWS3 methods is that they are constrained by each other's bottleneck. In our experiment, we use the simple KWS

Table 9: Detail Performance of KWS + PA

Model	Recall All	Recall Real	Recall Fake	False Alarm
KWS 1	2149/2458	1221/1714	572/ 744	49/21997
PA1	1221/1577	-	-	(31+56)/(890)
PA2	1492/1577	-	-	(44+158)/(890)
KWS 1+PA 1	1221/1714	-	-	(31+56)/(21997+774)
KWS 1+PA 2	1492/1714	-	-	(44+158)/(21997+774)

module and PA module compared with the current state-of-the-art approaches to make the comparison between different approaches more convincing. If some mistakes have been made in the KWS module and the PA model will further amplify the mistake. Further work can be done by using more complex models to improve each module and find innovative ways in model joint-training.

In conclusion, in this paper, we defined an emotional KWS task and proposed the dataset captured in the elevator scene. We investigated and compared the performance of the LVCSR KWS approach, the two-stage KWS + PA approach, and the KWS only classification approaches. We found that the small-footprint DNN based approach is more efficient and effective compared to the LVCSR Lattice Search Method. Considering the performance of detecting real single "jiu ming" word is still not very satisfying, we could trigger the SOS system with two positive "jiu ming" detection in a short period, which makes this approach more robust in the real application.

5 ACKNOWLEDGEMENT

This research was funded by Kunshan Government Research (KGR) Funding in AY 2020/2021.

REFERENCES

[n. d.]. ST-CMDS-20170001_1, Free ST Chinese Mandarin Corpus. https://www.openslr.org/38/.

Ltd Beijing DataTang Technology Co. 2006. aidatatang_200zh, a free Chinese Mandarin speech corpus. Retrieved Sep 1, 2021 from http://www.openslr.org/62/

Guoguo Chen, S. Khudanpur, Daniel Povey, J. Trmal, David Yarowsky, and Oguz Yilmaz. 2013. Quantifying the value of pronunciation lexicons for keyword search in lowresource languages. *2013 IEEE International Conference on Acoustics, Speech and Signal Processing* (2013), 8560–8564.

Guoguo Chen, Carolina Parada, and G. Heigold. 2014. Small-footprint keyword spotting using deep neural networks. *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (2014), 4087–4091. https://doi.org/10.1109/ICASSP.2014.6854370

K. Chlasta, K. Wołk, and I. Krejtz. 2019. Automated speech-based screening of depression using deep convolutional neural networks. *ArXiv* abs/1912.01115 (2019).

David Crystal. 2019. *PARALINGUISTICS*. De Gruyter Mouton, 265–296. https://doi.org/doi:10.1515/9783111659916-008

Najim Dehak, P. Kenny, R. Dehak, P. Dumouchel, and P. Ouellet. 2011a. Front-End Factor Analysis for Speaker Verification. *IEEE Transactions on Audio, Speech, and Language Processing* 19 (2011), 788–798.

Najim Dehak, P. Kenny, R. Dehak, P. Dumouchel, and P. Ouellet. 2011b. Front-End Factor Analysis for Speaker Verification. *IEEE Transactions on Audio, Speech, and Language Processing* 19 (2011), 788–798.

Theodoros Evgeniou and Massimiliano Pontil. 2004. Regularized multi–task learning. In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*. 109–117.

Kaiming He, X. Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep Residual Learning for Image Recognition. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2016), 770–778.

Xingyu Na Bengu Wu Hao Zheng Hui Bu, Jiayu Du. 2017. AIShell-1: An Open-Source Mandarin Speech Corpus and A Speech Recognition Baseline. In *Oriental COCOSDA 2017*. Submitted.

Tom Ko, Vijayaditya Peddinti, Daniel Povey, M. Seltzer, and S. Khudanpur. 2017. A study on data augmentation of reverberant speech for robust speech recognition. *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (2017), 5220–5224.

S. Lalitha, D. Geyasruti, R. Narayanan, and M. Shravani. 2015. Emotion Detection Using MFCC and Cepstrum Features. *Procedia Computer Science* 70 (2015), 29–35.

Ltd. Magic Data Technology Co. 2019. Retrieved Sep 1, 2021 from http://www.imagicdatatech.com/index.php/home/dataopensource/data_info/id/101

Soo Jin Park, G. Yeung, J. Kreiman, P. Keating, and A. Alwan. 2017. Using Voice Quality Features to Improve Short-Utterance, Text-Independent Speaker Verification Systems. In *INTERSPEECH*.

Daniel Povey, Gaofeng Cheng, Yiming Wang, Ke Li, Hainan Xu, Mahsa Yarmohammadi, and Sanjeev Khudanpur. 2018. Semi-Orthogonal Low-Rank Matrix Factorization for Deep Neural Networks.. In *Interspeech*. 3743–3747.

Daniel Povey, A. Ghoshal, Gilles Boulianne, L. Burget, O. Glembek, N. Goel, M. Hannemann, P. Motlícek, Y. Qian, Petr Schwarz, J. Silovský, G. Stemmer, and Karel Veselý. 2011. The Kaldi Speech Recognition Toolkit.

Ltd. Primewords Information Technology Co. 2018. Primewords Chinese Corpus Set 1. https://www.primewords.cn.

T. Sainath and Carolina Parada. 2015. Convolutional neural networks for small-footprint keyword spotting. In *INTERSPEECH*.

Björn Schuller, S. Steidl, A. Batliner, F. Burkhardt, L. Devillers, Christian A. Müller, and Shrikanth S. Narayanan. 2013. Paralinguistics in speech and language - State-of-the-art and the challenge. *Comput. Speech Lang.* 27 (2013), 4–39.

Changhao Shan, Junbo Zhang, Yujun Wang, and L. Xie. 2018. Attention-based End-to-End Models for Small-Footprint Keyword Spotting. In *INTERSPEECH*.

Yao Shi, Hui Bu, Xin Xu, Shaoji Zhang, and Ming Li. 2020. AISHELL-3: A Multi-speaker Mandarin TTS Corpus and the Baselines. *CoRR* abs/2010.11567 (2020). arXiv:2010.11567 https://arxiv.org/abs/2010.11567

David Snyder, Guoguo Chen, and Daniel Povey. 2015. MUSAN: A Music, Speech, and Noise Corpus. *CoRR* abs/1510.08484 (2015). arXiv:1510.08484 http://arxiv.org/abs/1510.08484

David Snyder, D. Garcia-Romero, Daniel Povey, and S. Khudanpur. 2017. Deep Neural Network Embeddings for Text-Independent Speaker Verification. In *INTERSPEECH*.

Ming Sun, A. Raju, G. Tucker, S. Panchapagesan, Gengshen Fu, Arindam Mandal, S. Matsoukas, N. Strom, and S. Vitaladevuni. 2016. Max-pooling loss training of long short-term memory networks for small-footprint keyword spotting. *2016 IEEE Spoken Language Technology Workshop (SLT)* (2016), 474–480.

Martin Vondra and R. Vích. 2008. Evaluation of Speech Emotion Classification Based on GMM and Data Fusion. In *COST 2102 Conference*.

Dong Wang and Xuewei Zhang. 2015. THCHS-30 : A Free Chinese Speech Corpus. *CoRR* abs/1512.01882 (2015). arXiv:1512.01882 http://arxiv.org/abs/1512.01882

Peng Yang, C. Leung, Lei Xie, B. Ma, and Haizhou Li. 2014. Intrinsic spectral analysis based on temporal context features for query-by-example spoken term detection. In *INTERSPEECH*.