Investigation of DNN-Based Keyword Spotting in Low Resource Environments

Kaixiang Shen, Meng Cai, Wei-Qiang Zhang, Yao Tian, and Jia Liu

Abstract—**Keyword Spotting is a challenging task aiming at detecting the predefined keywords in utterances. In the low resource environment such as little keyword templates and the lack of linguistic information, the detection performance is always unsatisfactory. In this paper, we focus on the low resource situation where every keyword only has about 40 templates and the linguistic information is unknown. We explore using deep neural networks for acoustic modeling. In addition, we investigate several techniques including transfer-learning, multilingual bottleneck features, balancing keyword filler data and data augmentation to address the low resource problem and improve the system's performance. Compared with a query-by-example baseline system, substantial performance improvement can be obtained with our proposed keyword spotting system with deep neural network (KWS-DNN) framework.**

Index Terms—**Keyword spotting, DNN, acoustic model.**

I. Introduction

Keyword Spotting aims at detecting given keywords in audio streams. It is often used on spoken document indexing and retrieval [1], spoken language understanding [2], and hands-free device wake-up interface [3]. For KWS task, there are three widely used methods: large vocabulary continuous speech recognition (LVCSR) based, query-by-example (QBE) based and keyword-filler based methods.

The LVCSR based systems [4]-[6] focus on generating rich lattices and effective keyword indexing researching, and often have higher accuracy than keyword-filler based systems when the training data and prior info is sufficient. It is very powerful to search a large database of audio content offline. But sometimes to build a LVCSR system is too difficult and expensive. Because it requires sufficient language resources, including hundreds of hours of transcription and a reliable pronunciation dictionary. In many practical scenarios, where the language for detection is a minority language or even the language is unknown, it is impracticable to build such an LVCSR system for keyword spotting.

The QBE based systems are often used in low resource environment. A typical QBE approach simply uses posteriorgrams as features and a dynamic time warping (DTW) algorithm is approached on this features to match keyword templates from test data [7]-[10]. It can be used in extra-low resource situation where every keyword only has little templates. However, the performance quickly becomes saturated as the number of keyword templates increases.

The keyword-filler based systems often train a discriminative detector to split test segments into two classes: keyword and filler. The keyword class contains all predefined keywords and the filler class contains one filler or multi-fillers which stand for silence, non-keyword filler and so on.

In a real applicative low resource scenario, we are able to get and transcript dozens of templates for each keyword rather than one or two templates. So in this paper, we focus on this situation. We build a QBE-based system as baseline and focus on our proposed keyword spotting system with deep neural network (KWS-DNN) system. To train such a system, what we need are only the utterances containing keywords and the corresponding start-time and end-time information of the keywords. So, it is very flexible and needs little cost to apply online.

To compensate for the lacking of training data, we have tried many methods. First, we use multilingual bottleneck (MBN) features to replace filter-bank (Fbank) features and try data augmentation method to "add data". We also try transfer leaning, which uses a well-trained deep neural network (DNN) model as a seed, and replace its output layer for our KWS mission. The results prove that the effect of the MBN feature and transfer learning method is very encouraging. However, the data augmentation method doesn't have a significant effect. Finally, recognition results of KWS-DNN system achieve huge improvement than baseline.

Details of our proposed KWS-DNN system are described in Section II. In Section III, we introduce the QBE-based baseline. Experimental setup and results are presented in Section IV. Finally, conclusions are summarized in Section V.

II. System Description

Like other machine learning tasks such as image identification and continuous speech recognition, the system performance of KWS relies heavily on training data and training methods. In LVCSR based system, an acoustic model can be well trained by hundreds of hours of transcription. The acoustic model can be a traditional gaussian mixture model or a neural network model. Using the well-trained acoustic model with a related language model, we can decode the test data and obtain rich lattice. Followed by an effective keyword research algorithm, the LVCSR based system can always have good performance. The LVCSR based system is a generalization task system, and decoding results are generated from the lattice which contains not only the pre-defined keywords but also the other words in dictionary. The

Manuscript received February 29, 2016; revised April 12, 2016. This work was supported by the National Natural Science Foundation of China under Grant No. 61273268, No. 61370034, and No. 61403224.

The authors are with the Department of Electronic Engineering, Tsinghua University, China (e-mail: skx13@mails.tsinghua.edu.cn, cai-m10@mails.tsinghua.edu.cn, wqzhang@tsinghua.edu.cn, chinaty188@163.com, ljia@tsinghua.edu.cn).

doi: 10.18178/ijfcc.2016.5.2.458

keywords are treated equally with other words, so once the lattice is generated we can search any word in the dictionary if we want. But in a more common and real application scenario, it is very difficult and unrealistic to get enough data to train an phone-based acoustic model and language model for the LVCSR system.

In this paper, we just focus on the low resource environment, and we assume that getting a few training utterances containing keywords and their transcription info are practicable. We set the 40 as the number of each keyword's templates. Namely each keyword has about 40 templates contained in the training utterances. It turns out to be reasonable and suitable.

As for keyword-filler based systems [11]-[14], each keyword or filler unit has a corresponding generative GMM-HMM model. The keyword HMMs are represented by sub-word models such as mono-phone or tri-phone models and trained from a big transcribed dataset. And Filler HMMs are trained for absorbing not-keyword audio segments. There are several choices of filler models: single filler the set of mono-phone or tri-phone, words, and even a full LVCSR system [15]. At runtime, the system builds a Viterbi decoder to give recognition results. But in this low resource environment, it is always unsatisfactory to obtain a well-trained HMM for each keyword within about 40 templates. Besides the Viterbi decoder need a little runtime computation.

The QBE based method can be used in our low resource environment, but may have bottleneck and poor potential for further performance promotion. Because the QBE system is just taking an averaging among the templates when processing the multi-query situation. In this way, too much useful information contained in different templates is discarded.

We use the specific data to train a acoustic DNN model followed by a posterior handling method producing recognition results which is similar to the system described in [3]. We also investigate several techniques including transfer-learning, multilingual bottleneck features, balancing keyword filler data and data augmentation to address the low resource problem and improve the system's performance. Details are presented in the following section.

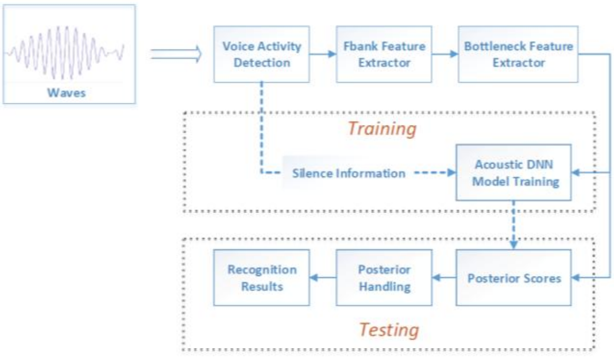


Fig. 1. KWS-DNN system.

A. Basic KWS-DNN System

In recent years, neural network has been used for KWS for acoustic modeling and feature extracting [3], [16]. In [3], a well-trained DNN followed with a posterior confidence score handling module is built to predict the keywords and has a substantial performance. In their work, they use a big training dataset called *VS data*. The *VS data* contains 3,000 hours of English training data and each keyword has about 2.3K templates. With the big training dataset and effective DNN acoustic modeling, this DNN-based framework outperforms the standard GMM-HMM based system.

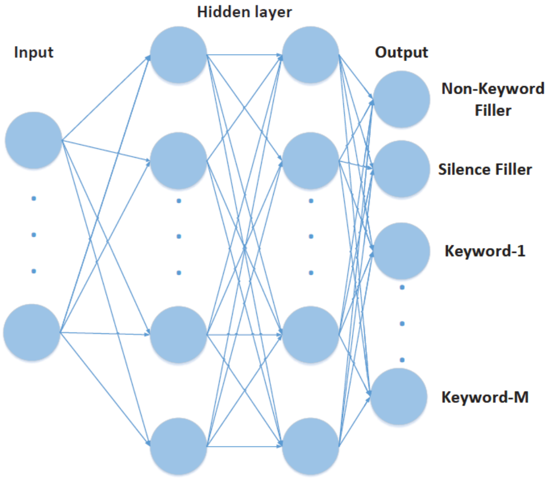


Fig. 2. Topology of the acoustic DNN.

Though we don't have much training data, the using of neural networks is inspiring. We build our own keyword-filler based DNN KWS system and we call it KWS-DNN system as shown in Fig.1. We use the DNN instead of the conventional GMM for acoustic modeling. The topology of the acoustic DNN is shown in Fig.2, each keyword has one corresponding unit in the output layer of the DNN model, and we add some fillers for absorbing noise and non-keyword utterance segments. A posterior handling is followed to give recognition results which is the same as the posterior handling modules described in [3]. We also try some methods for further performance improvement. We will firstly describe our system from training and testing.

At training, the inputs are utterances which contain keyword-fillers and their transcriptions. The transcriptions give the keyword boundary information and are used for keywords labeling. Before extracting utterance features, we implement a Voice Activity Detection (VAD) module to get boundaries of silence. The other parts of the utterance are labeled as non-keyword fillers. The Fbank feature is firstly extracted. With the Fbank features as inputs, the Bottleneck-Extractor produces bottleneck features. The Bottleneck-Extractor actually is a well-trained multi-layer forward propagation DNN acoustic model. With the training labels and bottleneck features, an acoustic DNN model for KWS is trained.

At testing, the utterance is firstly processed by the same VAD module, and only active voice regions will be processed. Acoustic posterior score is computed by this DNN model from bottleneck feature for every frame. A posterior handling module is followed which combines the label posteriors into a confidence score to process recognition results. The recognition results contain the keywords and fillers, as well as their time stamp and confidence scores.

B. Feature Extractor

There are several kinds of features that can be used: hand-designed features such as Fbank feature, PLP features, MFCC features, LPC features and data-driven features such as bottleneck features and LSTM features. Hand-designed features give the basic descriptions of the utterances from acoustic aspect. The LSTM feature extractor extracts a fixed dimensionality feature from a variable length audio segment [16]. Bottleneck features generated by a multi-layer perceptron (MLP) can be considered as a non-linear feature transformation and dimensionality reduction technique. They can extract useful information from multiple frames of the acoustic features and have been proved effective in improving the accuracy of ASR [17]. Motivated by this, we use multilingual bottleneck features for our KWS task. We believe that the robust feature which contains more useful information for phoneme classification can give performance improvement for our KWS task.

C. Balance Keyword and Filler Data

A highly unbalanced dataset is always a challenge for neural network training. In our task, it is also a problem where negative frames of filler segments are always hundreds of times over positive keywords frames.

One way to balance fillers and keywords is to re-sample the training examples and keep the filler examples the same magnitude with the keyword examples. It's called re-sampling method. Down-sampling the fillers is a solution, but only using down-sampling isn't suitable. If we down-sample fillers, there will be little frames for training and we'll lose too much informative instances. Over-sampling is another re-sampling method which increases the number of keyword instances by over-sampling them. In this way, no information is lost from the training samples [18]. However, the minority keyword instances are over-represented in the training set, and moreover, adding training instances means increasing training time.

Another way is to use weighted example method. This method gives positive and negative examples different weights to train our acoustic DNN model. While we train the DNN, the activation function is sigmoid and the related error function is common cross-entropy (CE), defined as Eq.1. We give different weighted CE error function shown in Eq.2.

$$C = -\sum_{\mathrm{j}} d_j . \log p_j \tag{1}$$

$$C_w = -w\sum_{j} d_j . \log p_j \tag{2}$$

where p_j is the normalized output probability for state j, and d_j is the label value. When the frame is labeled for state j, the value of d_j is 1 otherwise is 0. Value w is the weight value. The weight value of positive keyword frames is 1, and weight value of negative filler frames is less than 1.

Multiplied with each frame's back-propagate errors, the weight value can give different contribution to back-propagate errors of positive and negative instances, which means that when the training instance is negative it has less weight and opportunity to update the parameters of DNN model. Using the weighted example method, we can include more filler instances for training.

In our experiment, we combine the down-sampling and weighted example methods. We firstly down-sample the negative examples and just keep fillers near positive keywords and try to control the negative examples that are dozens of times over positive examples, and then use weighted example method to train DNN.

D. Transfer Leaning and Data Augmentation

We also try transfer leaning and data augmentation methods. The transfer learning here refers to the situation where the KWS DNN parameters are initialized with the corresponding parameters of an existing well-trained network, and are not trained from [19], [20]. We firstly train a DNN model for speech recognition with a suitable topology using a big multilingual dataset, and then use this well-trained DNN model as a seed to initialize the hidden layers of our KWS network. We replace its output layer with our task-related output layer, and then all layers are updated during re-training. In this way, the hidden layers has potentiality to learn a better and more robust feature representation by exploiting larger amounts of data and avoiding bad local optima [19].

We also try data augmentation method. Namely we use transforms of the data as the input while preserving the labels. We explore two data argumentation methods including adding noise and vocal tract length perturbation (VTLP) data [21]. Four kinds of noise types (babble noise, pink noise, subway noise and white noise) and four VTLP warp factors (0.92, 0.96, 1.04 and 1.08) are used to generate augmentation dataset. We randomly choose part of augmentation dataset along with the raw dataset to be the whole training dataset. In our experiment, transfer leaning method is effective, but data augmentation method doesn't gain improvement.

III. QBE-Based Baseline

In the low resource environment, QBE-based approach is a simple and feasible solution for KWS task. It has advantage in extra-low resource environment where every keyword only has less than 5 templates. We use the QBE-based system as our baseline.

Our baseline is similar as the system described in [22]. At training, phone posteriorgram feature vectors of keyword queries are extracted at frame-level. If a keyword has multi-queries, we align their feature vectors into the longest query of this keyword and get an averaged feature vector. And in running time, input audio document is firstly processed by a VAD module, and only active audio segments are retained for extracting features. A distance matrix is computed between audio segments and keywords, followed by a DTW matching procedure. A confidence score is then obtained from the DTW alignment cost.

We train a DNN model for phoneme posteriorgram features extracting. The model has 5 hidden layers with 1024 nodes for each layer and training data contains 700 hours of Mandarin Chinese and 700 hours of English data. The last layer of the DNN model contains 137 states for 39 English monophone, 96 Mandarin Chinese, silence and short-pause (sp). We use sigmoid as activation function and use stochastic gradient descent (SGD) to optimize the CE target.

IV. Database and Results

Use In this section, we will present our experimental settings and results.

Our goal is to build a light-weighted KWS system under the low resource environment with little training data and no language information. To simulate this scenario, we use a small dataset collected by the Speech and Audio Technology Laboratory of Tsinghua University (THU-SATLab). This dataset contains about 40 hours of audios of Chinese. We choose 30 keywords and every keyword only has about 40 queries. The testing audio is about 10 hours.

We use F_1 score as measurement of system performance and it is defined as Eq.3:

$$F_1 = 2\frac{P_P.P_r}{P_p + P_r} \tag{3}$$

where P_p stands for precision probability which is the value of the number of correct positive results divided by the number of all positive results, and P_r stands for recall probability which is the value of the number of correct positive results divided by the number of positive results that should have been returned. The F_1 score is higher and performance is better.

We firstly compare filter-bank (Fbank) with bottleneck feature. The results are listed in Tab.I. The Fbank feature used is a 120-dimensional feature vector containing 40-dimensional Mel filter-bank feature and their first- and second-order derivatives. The Bottleneck feature is a 256-dimensional vector extracted from a Bottleneck-DNN trained with 700 hours of English and 700 hours of Mandarin Chinese audios. In our experiment, we don't train a new phoneme posteriorgram feature extractor with bottleneck feature as input. Because we think the 1400-hour well-trained phoneme posteriorgram feature extractor has the same effect for extracting useful phoneme classification information while training. KWS-DNN Down-Sampling in Table I, means that we down-sample the negative fillers and make sure they have the same magnitude frames with positive keywords. The acoustic DNN has 2 hidden layers with 256 nodes. The context window size is 11 and error function is CE.

TABLE I: F1 Values (%) by Fbank and Bottleneck Features

Systems	Fbank	Bottleneck
Baseline	5.76	-
KWS-DNN Down-Sampling	8.27	26.86

TABLE II: F1 Values (%) for KWS-DNN System

Systems	Not use seed	Use seed
Down-Sampling and Weighted examples	33.82	37.87
Data Augmentation	33.15	37.81

As shown in Table I, for Fbank feature, KWS-DNN Down-Sampling system's $F1$ value is better than QBE baseline. We think the reason for the performance improvement is that KWS-DNN system uses information of multi-queries more effective than baseline and QBE baseline loses much useful information while doing multi-queries averaging.

The results of transfer leaning, balancing keyword and filler data, and data augmentation methods are shown in Table II. The seed DNN model for transfer leaning described in section above has 2 hidden layers and each hidden layer has 256 nodes. For balancing keyword and filler data, we down-sample the negative filler examples and set the ratio of fill frames to keyword frames to about 15. The weight value *w* for weighted example method is 0.02. In the Table II, data augmentation is used on the base of down-sampling and weighted example method.

The balancing keyword-filler method as down-sampling and weighted examples is proved effective in our task. It has a 25.9% relative improvement. We think the reason is that it includes more negative data for training and has a better absorption of different kinds of noises. The transfer learning is also important for performance improvement. The highest F_1 score (37.87%) is obtained with transfer learning and balanced data methods. As to the data augmentation method, the results are even worse. We think a reasonable explanation is that no additional information is added in the system.

V. Conclusion

In this paper, we have proposed a KWS-DNN keyword spotting system. Experimental results show that the proposed framework outperforms QBE based system. Different kinds of methods are proposed to further improve the system's performance and we demonstrate that the using of bottleneck feature, down-sampling and weighted examples and transfer learning can improve the system's performance in our task.

Acknowledgment

We would like to thank Tingting Cheng for her devotion and help for the writing of this paper.

References

[1] J. Makhoul, F. Kubala, T. Leek, D. Liu, L. Nguyen, R. Schwartz, and A. Srivastava, "Speech and language technologies for audio indexing and retrieval," in *Proc. the IEEE*, vol. 88, no. 8, pp. 1338–1353, Aug. 2000.

[2] B.-H. Juang and S. Furui, "Automatic recognition and understanding of spoken language — A first step toward natural human-machine communication," in *Proc. the IEEE*, vol. 88, no. 8, pp. 1142–1165, Aug. 2000.

[3] G. Chen, C. Parada, and G. Heigold, "Small-footprint keyword spotting using deep neural networks," in *Proc. 2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, May 2014, pp. 4087–4091.

[4] S. Parlak and M. Saraclar, "Spoken term detection for Turkish broadcast news," in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing*, March 2008, pp. 5244–5247.

[5] J. Mamou, B. Ramabhadran, and O. Siohan, "Vocabulary independent spoken term detection," in *Proc. the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, New York, NY, USA: ACM, 2007, pp. 615–622.

[6] D. R. H. Miller, M. Kleber, C. L. Kao, O. Kimball, T. Colthurst, S. A. Lowe, R. M. Schwartz, and H. Gish, "Rapid and accurate spoken term detection," *in Proc. Interspeech*, pp. 314–317, 2007.

[7] B. Song, W. Zhang, M. Cai, J. Liu, and M. T. Johnson, "Query-byexample spoken term detection based on phonetic posteriorgram query by example spoken term detection based on phonetic posteriorgram," in *Proc. International Conference on Education, Management and Computing Technology (ICEMCT-15)*, 2015.

[8] L. J. Rodrłguez-Fuentes, A. Varona, M. Peagarikano, G. Bordel, and M. Dłez, "High-performance query-by-example spoken term detection on the sws 2013 evaluation," in *Proc. ICASSP*, 2014, pp. 7819–7823.

[9] T. Hazen, W. Shen, and C. White, "Query-by-example spoken term detection using phonetic posteriorgram templates," in *Proc. IEEE Workshop on Automatic Speech Recognition Understanding*, Nov 2009, pp. 421–426.

[10] I. Szoke, L. Burget, F. Grezl, J. H. Cernocky, and L.Ondel, "Calibration and fusion of query-by-example systems ł but sws 2013," in *Proc. 2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2014, pp. 7849–7853.

[11] R. Rose and D. Paul, "A hidden markov model based keyword recognition system," in *Proc. 1990 International Conference on Acoustics, Speech, and Signal Processing*, Apr. 1990, pp. 129–132, vol. 1.

[12] J. R. Rohlicek, W. Russell, S. Roukos, and H. Gish, "Continuous hidden markov modeling for speaker independent word spotting," in *Proc. International Conference on Acoustics, Speech, and Signal Processing*, 1989, pp. 627–630.

[13] J. Wilpon, L. Miller, and P. Modi, "Improvements and applications for key word recognition using hidden markov modeling techniques," in *Proc. 1991 International Conference on Acoustics, Speech, and Signal Processing*, Apr. 1991, pp. 309–312.

[14] G. David, K. Joseph, and B. Samy, "Discriminative keyword spotting," *Speech Communication*, vol. 51, no. 4, pp. 317–329, 2009.

[15] M. Weintraub, "Keyword-spotting using sri's decipher large-vocabulary speech-recognition system," in *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing*, 1993, pp. 463–466.

[16] G. Chen, C. Parada, and T. Sainath, "Query-by-example keyword spotting using long short-term memory networks," in *Proc. 2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, April 2015, pp. 5236–5240.

[17] D. Yu and M. L. Seltzer, "Improved bottleneck features using pretrained deep neural networks," *Interspeech*, pp. 237–240, 2011.

[18] G. H. Nguyen, A. Bouzerdoum, and S. L. Phung, "A supervised learning approach for imbalanced data sets," in *Proc. 19th International Conference on Pattern Recognition*, 2008, pp. 1–4.

[19] R. Caruana, "Multitask learning," *Machine Learning*, vol. 28, no. 1, pp. 41–75, 1997.

[20] G. Heigold, V. Vanhoucke, A. Senior, P. Nguyen, M. Ranzato, M. Devin, and J. Dean, "Multilingual acoustic models using distributed deep neural networks," in *Proc. International Conference on Acoustics, Speech, and Signal Processing*, 2013, pp. 8619–8623.

[21] M. Cai, Z. Lv, C. Lu, J. Kang, L. Hui, Z. Zhang, and J. Liu, "High performance swahili keyword search with very limited language pack: The thuee system for the openkws15 evaluation," in *Proc. IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*, Dec. 2015, pp. 215–222.

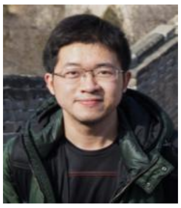
[22] B. Song, W.-Q. Zhang, M. Cai, J. Liu, and M. T. Johnson, "Query-by-example spoken term detection based on phonetic posteriorgram," in *Proc. International Conference on Education, Management and Computing Technology (ICEMCT-15)*, vol. 10, Jun. 2015, pp. 1255–1260.

Kaixiang Shen was born in Hunan, China, in 1991. He received the B.S degree in communication engineering from Minzu University of China in 2013. Currently, he is a master student in the Department of Electronic Engineering, Tsinghua University. His research interests include speech recognition and keyword spotting.

Meng Cai received his B.S. degree in Beijing Institute of Technology in 2010, and his Ph.D. degree in Tsinghua University in 2016, both in Electronic Engineering. He is now an associate researcher at Microsoft Research Asia. His research interests include speech recognition, acoustic modeling and deep learning.

Wei-Qiang Zhang received the B.S. degree in applied physics from China University of Petroleum in 2002, the M.S. degree in communication and information systems from Beijing Institute of Technology in 2005, and the Ph.D. degree in information and communication engineering from Tsinghua University in 2009.

He is an associate professor in the Department of Electronic Engineering, Tsinghua University. His research interests are in the area of speech and signal processing, machine learning and statistical pattern recognition.



Yao Tian received his B.S. degree in University of Electronic Science and Technology of China in 2011. He is currently a Ph.D. student in the Department of Electronic Engineering, Tsinghua University. His research interests include speech recognition, speaker verification and deep learning.

Jia Liu received the B.S., M.S., and Ph.D. degrees in communication and electronic systems from Tsinghua University, Beijing, China, in 1983, 1986, and 1990, respectively. He worked at the Remote Sensing Satellite Ground Station, Chinese Academy of Sciences, after the Ph.D. degree and worked as a royal society visiting scientist at the Cambridge University Engineering Department, Cambridge, U.K., from 1992 to 1994.

He is now a professor in the Department of Electronic Engineering, Tsinghua University. His research fields include speech recognition, speaker recognition, language recognition, expressive speech synthesis, speech coding, and spoken language understanding.