End-to-end Keywords Spotting Based on Connectionist Temporal Classification for Mandarin

Ye Bai[1, 3] *Jiangyan Yi*[1, 3] *, Hao Ni*[1, 3]*, Zhengqi Wen*[1]*, Bin Liu*[1]*, Ya Li*[1]*, Jianhua Tao*[1, 2, 3]

[1]National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences, Beijing 100190, China
[2]CAS Center for Excellence in Brain Science and Intelligence Technology, Chinese Academy of Sciences, Beijing 100190, China
[3]School of Computer and Control Engineering, University of Chinese Academy of Sciences, Beijing 100049, China

baiye2016@ia.ac.cn, {jiangyan.yi, hao.ni, zqwen, liubin, yli, jhtao}@nlpr.ia.ac.cn

Abstract

Traditional hybrid DNN-HMM based ASR system for keywords spotting which models HMM states are not flexible to optimize for a specific language. In this paper, we construct an end-to-end acoustic model based ASR for keywords spotting in Mandarin. This model is constructed by LSTM-RNN and trained with objective measure of connectionist temporal classification. The input of the network is feature sequences, and the output the probabilities of the initials and finals of Mandarin syllables. Compared with hybrid based ASR systems, the end-to-end system achieves a significant improvement of 6.32% on ATWV relatively. The best result of our system is ATWV 0.8310 on RASC863 data set. The proposed CTC based method applies to KWS in a specific language.

Index Terms: keywords spotting, LSTM-RNN, connectionist temporal classification, end-to-end

1. Introduction

Keywords spotting (KWS) is to detect a pre-defined set of spoken terms in the given unconstrained speech [1]. It has been widely used in voice-dialing, call center, voice monitoring, voice controlling, speech retrieval, and so on. Among these applications, there are two main types of approaches applied in KWS. One is the supervised approaches, for example, the Large Vocabulary Continuous Speech Recognition (LVCSR) based approach [2], and the other is unsupervised approaches, such as template matching [3]. Some methods, such as filler model [4] and DNN [5], need to be retrained after changing keywords list. In non-specific tasks for the KWS, the LVCSR based approach is widely used since that it does not require any prior knowledge about speech for searching the keywords. It is flexible to change keywords according to users' requirement. In this paper, we focus on KWS based on LVCSR.

In the LVCSR based approach, speech is firstly converted into a form of text data structures, and then an inverted index is constructed for searching the users' keywords. Because 1-best word level output of the LVCSR is not entirely accurate, it will affect the performance of KWS. Structures which can provide more candidate results for searching keywords are proposed, such as position specific posterior lattice (PSLP) [6]. The raw text structures contain redundant words, and the keywords might be spotted in them.

The LVCSR system is firstly constructed which includes an acoustic model and a language model. The acoustic model generates the posterior probability given the input acoustic features which includes a hidden Markov model (HMM) and a Gaussian mixture model (GMM) [7] or deep neural network (DNN) [8]. The HMM is used to describe relation between an acoustic feature sequence and a state sequence to model a phone, and the GMM or DNN is used to model relation between an acoustic feature and a HMM state. The language model trained by the large-scaled corpus is further adopted to construct the weighted finite-state transducer (WFST) [9, 10] for decoding.

However, building such a LVCSR system is complicated. The construction of acoustic model is divided into several stages. State level model is constructed without actual meaning in phonetics. It is difficult to bring in knowledge of phonetics for specific language to acoustic model. So it is not convenient to improve performance of keywords spotting for a specific language such as Mandarin.

Recently, end-to-end based acoustic model is proposed for the LVCSR, such as the connectionist temporal classification (CTC) [11] and attention based model [12]. CTC is a direct method for sequence labelling tasks with recurrent neural network model. It can simplify the architecture of LVCSR with a single recurrent neural network (RNN) [13]. Without modelling the HMM states, CTC could generate the posterior probability for the phonetic elements given the input acoustic features, such as phones, syllables or characters.

In this paper, we construct our keywords spotting system based on CTC for Mandarin. We investigate two kinds of features, mel-frequency cepstrum coefficient (MFCC) and mel-scale filter bank (FBANK) to train the RNN respectively. The model is constructed for the initials and finals of Mandarin syllables. The experiments are carried out to compare with the traditional DNN-HMM based acoustic models. The experimental results show the advantage of the proposed method.

initials
finals

The rest of this paper is organized as follows: In section 2, we describe the structure of our ASR system based on connectionist temporal classification. Acoustic model training method is introduced. The search algorithm is introduced in section 3. Then section 4 describes experimental setup and results. Finally, conclusions and future work are presented in section 5.

2. CTC Based ASR

2.1. CTC Based Acoustic Model

The structure of acoustic model in typical ASR systems can be represented as two levels: HMM level is composed of a set of clustered states, and state's output distribution level is represented by GMM or DNN. The CTC based acoustic model unifies two level structures to a single RNN based framework.

The main problem in speech recognition is to convert an acoustic feature sequence to a character sequence. But the relation between these two sequences cannot be modeled directly by RNN. Because the length of character sequence is often shorter than acoustic feature sequence, when the labels created by RNN are corresponded one to one with input sequence. CTC is proposed to solve this problem. The main idea is to add a blank symbol to the set of labels and to label with RNN. At last, remove the extra blank symbols and repeated symbols [11]. The model is described as follows.

For a given vector sequence of length T and a set of labels L, define a function mapping input M-dimensional vector sequence to N-dimensional output vector sequence:

$$\mathbf{Y} = \mathrm{F}(\mathbf{X}) \tag{1}$$

where, $\mathbf{X} = (\boldsymbol{x}_1, \boldsymbol{x}_2, \boldsymbol{x}_3, \dots, \boldsymbol{x}_\mathrm{T})$ is the input vector sequence, and $\mathbf{Y} = (\boldsymbol{y}_1, \boldsymbol{y}_2, \boldsymbol{y}_3, \dots, \boldsymbol{y}_\mathrm{U})$ is output vector sequence of length $U \leq T$.

Every component of $\mathbf{y}$ represents the probability of occurrence of each label. Let y_k^t be an output component of unit k at time t, π be a candidate path. Assuming each probability of output symbol is independent, the probability of a path is defined as follows:

$$\mathrm{P}(\pi|\mathbf{X}) = \prod_{t=1}^{T} y_{\pi_t}^t, \forall \pi \in L' \tag{2}$$

where $L' = L \cup \{blank\}$.

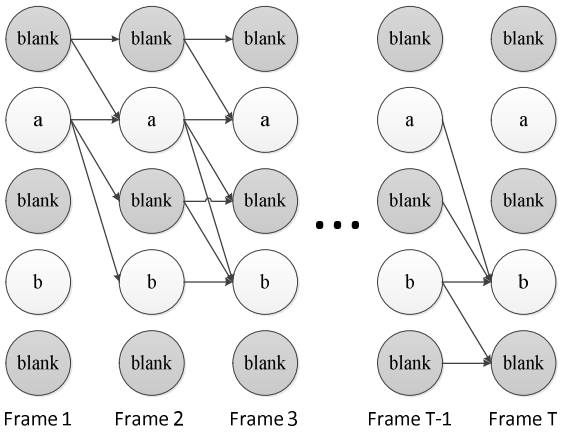

Figure 1: *Trellis of the labelling "ab"*

The final result we need is the sequence which does not have blank symbol. A lot of sequences generated by equation (1) can map to sequence which does not have blank symbol by removing blanks and repeated labels. For example, for sequence "ab", "-aa-b" or "-aa-bb-" can be candidate sequences. Defining a many-to-one map $B: L'^T \rightarrow L^U$, and B^{-1} is the inverse of B, the conditional probability of a labelling $\mathbf{l} \in L^U$ can be represented as

$$\mathrm{P}(\mathbf{l}|\mathbf{X}) = \sum_{\pi \in B^{-1}(\mathbf{l})} \mathrm{P}(\pi|\mathbf{X}) \tag{3}$$

The sum is intractable to calculate. It is effective to calculate the sum by bringing in Forward-Backward algorithm in hidden Markov model. First, represent all the possible CTC paths as a trellis. Add blanks to the beginning and the end of the sequence, and insert blanks between each symbols pair of original sequence. So the length of candidate sequence is *2U+1*. All the path on the trellis from the upper-left corner to lower-right corner can be mapped to the result path.

Define forward probability as the total probability of all CTC paths ending up with label π_u at frame t:

$$\alpha_t(\pi_u) = \sum_{\mathrm{B}(\pi_{1:t})=\mathbf{l}_{1:t}} \prod_{\tau=1}^{t} y_{\pi_\tau}^t \tag{4}$$

The $\alpha_t(\pi_u)$ can be calculated iteratively from $\alpha_{t-1}(\pi_u)$ and $\alpha_{t-1}(\pi_{u-1})$.

Also define a backward probability as follows:

$$\beta_t(\pi_u) = \sum_{\mathrm{B}(\pi_{1:t})=\mathbf{l}_{t:T}} \prod_{\tau=t}^{T} y_{\pi_\tau}^t \tag{5}$$

So the likelihood of the final result can be represented as

$$\mathrm{P}(\mathbf{l}|\mathbf{X}) = \sum_{u=1}^{2U+1} \alpha_t(\pi_u)\, \beta_t(\pi_u) \tag{6}$$

The partial derivative of the objective $\ln \mathrm{P}(\mathbf{l}|\mathbf{X})$ corresponded to the component y_k^t is

$$\frac{\partial \ln \mathrm{P}(\mathbf{l}|\mathbf{X})}{\partial y_k^t} = \frac{1}{\mathrm{P}(\mathbf{l}|\mathbf{X})} \frac{1}{y_k^t} \sum_{\pi_u \in \{u|\mathbf{l}_u=k\}} \alpha_t(\pi_u)\, \beta_t(\pi_u) \tag{7}$$

So the backpropagation algorithm can be used by propagate the gradient through the softmax layer.

2.2. Decoding for CTC Based ASR

The decoding method which combines acoustic cost and language model cost is based on weighted finite state transducer (WFST) [13]. A Token WFST maps CTC sequence to phone sequence. A Lexicon WFST maps phone sequence to words sequence. A Grammar WFST is a weighted finite state accepter which save language score on arcs. The final search graph is constructed by composing the three WFSTs. The formula of construction is

$$\mathrm{S} = \mathrm{T} \circ \min(\det(\mathrm{L} \circ \mathrm{G})) \quad \text{TLG} \tag{8}$$

Where "$\circ$" means composition, "*det*" means determinization, and "*min*" means minimization. These are basic operations of WFST.

To provide more candidate results for keywords spotting, the decoding results are saved as lattices. The keywords are searched in the lattices.

3. Keywords Spotting Based on CTC

The diagram of the system is shown in Figure 2. The front-end of keywords spotting system is an ASR system. Then the candidate results of ASR will be converted to an index for searching keywords.

The search index is constructed with timed factor transducer algorithm [14]. Timed factor transducer is a kind of weighted finite state transducer which accepts all substrings of any path in the lattice. The weight of an arc of a timed factor transducer is a three tuple which saves score, start time, and end time. The index of a given speech is constructed by taking the union of all the timed factor transducers.

The lattice is preprocessed before construction. The time steps of every state in lattice can be recorded by traversing after topological sort. And then the period of every arc is obtained. The arcs are clustered according to input labels and overlapping periods. First, sort arcs in terms of end time steps. Then find the largest non-overlapping (start time, end time) pairs as cluster heads. Finally, assign cluster ID to the rest of arcs. The input of factor transducers are input labels, and the output labels are cluster IDs.

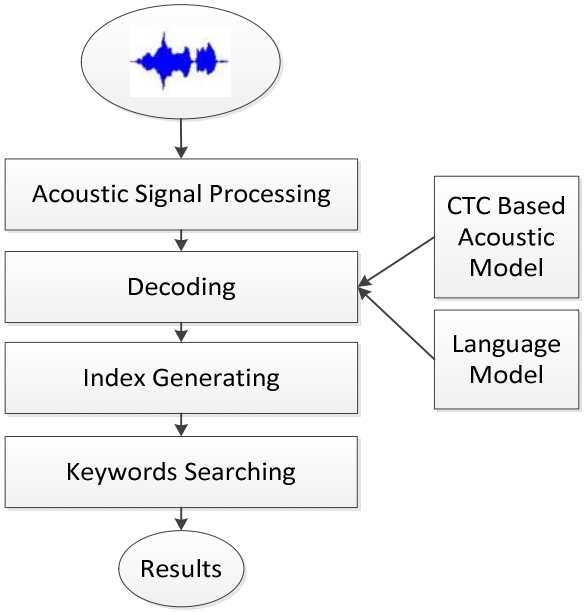

Figure 2: *Illustration of proposed KWS system*

Let $B_i = (\Sigma, \Delta, Q_i, I_i, F_i, E_i, \lambda_i, \rho_i)$ denotes a transducer over the log semiring after preprocessing. Where Σ is the input alphabet, Δ is the output alphabet, Q is a set of states, I is the set of initial states, F is the set of final states, E is the set of arcs of the transducer, λ is the initial weight function, and ρ is the final weight function. The weight of B_i represents occurrence probability $P_i(x, y)$ for each string pair $(\mathrm{x}, \mathrm{y}) \in \Sigma^* \times \Delta^*$. $P_i(x, y)$ is the sum of the probabilities of all successfully paths in B_i where (x, y) is a factor. $P_i(x, y)$ can be computed using Forward-backward algorithm. Let $d[q]$ be the total probability from first state to state q, and $f[q]$ be the total probability from state q to final state. Let $t_i^s(x, y)$ and $t_i^e(x, y)$ denotes start time and end time of factor (x, y). Then construct a transducer mapping every factor to a 3-tuple $\left(P_i(x, y), t_i^s(x, y), t_i^e(x, y)\right)$.

First, set the weight of every arc $\{w\}$ as $\{w, 0, 0\}$. Create a new initial state s, and create a new final state e. For each original state, create two new arcs: an initial arc $(s, \varepsilon, \varepsilon, \{d[q], t_i[q], 0\}, q)$, and a final arc $(q, \varepsilon, i, \{f[q], 0, t_i[q]\}, e)$. Then merge the paths which have similar factor. The transducer is optimized using minimization and determinization. The index is constructed by union all the transducers.

Searching is divided into two steps. First, compiling the query string to an linear finite state acceptor. And then compose the acceptor with the index. The time information of where the keywords occur can be obtained by projecting the WFST.

"Proxies" method is used to handle OOV queries [15]. Proxy word is an IV word whose pronunciation is similar to the given OOV word. The acceptor of a proxy word is generated as:

$$\mathrm{K}' = \mathrm{Project}(\mathrm{ShortestPath}(\mathrm{K} \circ \mathrm{L}_2 \circ \mathrm{L} \circ \mathrm{L}_1^{-1})) \quad (7)$$

where K is the acceptor of the given OOV word, L_2 is a WFST mapping a word to its pronunciation which obtained by Sequitur grapheme-to-phoneme tool [16], and L_1 is the lexicon WFST of the LVCSR system. E is a WFST which maps a phone sequence to another phone sequence with edit-distance metric. The ShortestPath operation retains shortest N paths as proxies. At last, the acceptor of proxies generated by Project operation. And then, the OOV word can be searched as an IV word with its proxy.

4. Experiments

4.1. Experimental Setup

The experiments are implemented using open source toolkit Kaldi [17] and EESEN [13]. The proposed network used in training the CTC based acoustic model is constructed by LSTM-RNN. The network consists of four unidirectional LSTM layers, each layer has 320 cells. Two kinds of input are tried to test the effect. One is 120-dimentional input generated from 40-dimensional log mel-frequency filter bank feature vector with delta and double deltas. The other is 39-dimensional which is generated from 13-dimensional mel-frequency cepstrum coefficients with delta and double deltas. The output is a 242-dimensional vector represents probabilities of 61 the initials and finals of Mandarin syllables, 175 disambiguation symbols, 5 auxiliary symbols, and a blank symbol. The network is trained with backpropagation through time (BPTT) [18]. The initial learning rate is 0.00002. The models are trained on RASC863: 863 annotated 4 regional accent speech corpora [19]. The corpora contains 250 hours of speech in Mandarin. The speech is sampled at 16kHZ.

We use data extracted from RASC863 to test the effectiveness of our proposed KWS system. The test set contains 20 hours of speech data which has not been included in the training data. The keyword list which contains 4253 keywords is generated randomly from labelling text. The number of out-of-vocabulary (OOV) keywords is 16. Because the OOV keywords are too few to provide a convincing result, we mainly focus on in-vocabulary (IV) KWS.

The language model is trigram which is trained by open source toolkit SRILM [20]. The text data which contains 30792 words is self-collected. The size of final WFST searching graph is 118MB.

The metric to measure the effectiveness of KWS is term-weighted value (TWV) [21]. It is an overall merit of detection performance with the weighted sum of the term-weighted probability of missed detection and the term-weighted probability of false alarms.

$$\mathrm{TWV}(\theta) = 1 - [P_{Miss}(\theta) + \beta P_{FA}(\theta)] \quad (9)$$

where θ is a threshold to determine if the system-detected keyword is scored. $P_{Miss}(\theta)$ is the frequency of missed detection and $P_{FA}(\theta)$ is the frequency of false alarms.

$$P_{Miss}(\theta) = \frac{1}{K}\sum_{kw=1}^{K} \frac{N_{Miss}(kw,\theta)}{N_{True}(kw)} \quad (10)$$

$$P_{FA}(\theta) = \frac{1}{K}\sum_{kw=1}^{K} \frac{N_{FA}(kw,\theta)}{N_{NT}(kw)} \quad (11)$$

where $N_{Miss}(kw, \theta)$ is the number of missed detection of the keyword kw for θ, $N_{FA}(kw, \theta)$ is the number of false alarms of the keyword kw for θ, $N_{True}(kw)$ is the number of reference occurrences of the keyword kw, and $N_{NT}(kw)$ is the number of non-target trials for keyword kw. β is a penalty coefficient which typically set as 999.9.

Actual TWV (ATWV) is an evaluation measure calculated by using the TWV for system occurrences with 'YES' hard decisions. Maximum term-weighted value (MTWV) also is used to measure spotting effect. The results of the experiment are evaluated by NIST F4DE evaluation tool [21].

4.2. ASR Experiment

We compare the effect of the CTC based acoustic model with DNN-HMM model in ASR. The input features for the DNN are FBANK. The DNN has 6 hidden layers, every layer has 1024

units. The model is sequence discriminatively trained using sMBR criterion [22]. The results are shown in Table 1.

Table 1. *Comparison of WER between baseline and CTC approach.*

Model	WER
DNN-HMM	7.12%
CTC(FBANK)	2.60%
CTC(MFCC)	2.06%

Compared with traditional DNN-HMM based ASR system, the WER of CTC based model decreases by 5.06%. The input of MFCC has the highest precision.

4.3. KWS Experiment

First we investigate the effect of two hyper-parameters which would influence ATWV of KWS: the width of decoding beam and the weight of acoustic cost. The two hyper-parameters are independent.

We investigate effect of beam on ATWV first. The result is shown in Figure 3. The experiment sets acoustic scale at 0.7.

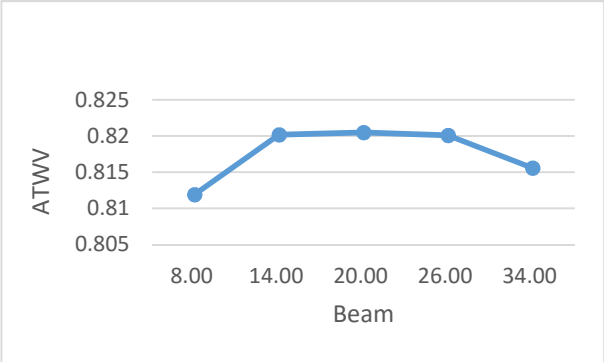

Figure 3: *ATWV versus beam*

The beam width influences the number of candidate sentences in a lattice. A higher beam provides more candidate words for KWS. On the other hand, because of the inaccuracy of the weight in lattice, it causes the increase of false alarms. Since too large beam causes increase of the size of the lattice, and it does not improve ATWV, we test 5 values of beam from 8 to 34. The result shows that ATWV increases from 8 to 14, and decreases from 26 to 34. The effect of changing beam from 14 to 26 is not obvious. The highest ATWV is at beam 20. So we set beam as 20 in the rest of experiments.

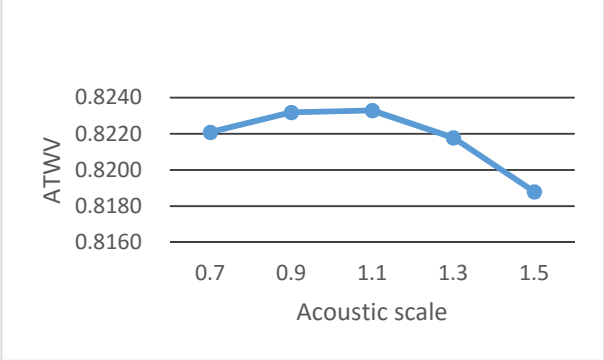

Figure 4: *ATWV versus acoustic scale*

We also investigate the effectiveness of weight of acoustic cost on ATWV. The ATWV is examined at 5 acoustic scales. The result is shown in Figure 4. The ATWV increases from acoustic scale 0.7 to 1.1, and then decreases. We set acoustic scale as 1.1 in the rest of the experiments. The acoustic scale is a parameter to balance the effect of acoustic model with language model. We consider that the weight of acoustic cost is more important than language model. Because in KWS task, the target is to find the appropriate word, not to recognize the whole sentence. The acoustic cost is more important for a single keyword. But the contextual occurrence information of the keywords in sentence is important to decrease false alarms. So the acoustic scale cannot be set too large.

The effort of KWS is shown in Table 2. The MFCC based CTC model has the highest ATWV and MTWV. The phone based CTC acoustic model with FBANK features gets Word Error Rate (WER) of 2.60%. And WER of phone based CTC model with MFCC inputs is 2.06%. ATWV of CTC model with MFCC inputs is 0.8310. Compared with DNN-HMM, the ATWV is improved relatively by 6.32%.

Table 2. Comparison of ATWV and MTWV between baseline and CTC approach.

Model	ATWV	MTWV
DNN-HMM	0.7816	0.7853
CTC(FBANK)	0.8225	0.8268
CTC(MFCC)	0.8310	0.8328

Traditional ASR system is divided to several parts, and every part has its own training objective. The end-to-end model unifies the whole system, and models the initials and finals of Mandarin syllables directly with RNN. It avoids inconsistency of objectives in multi-level system. That is effective to improve WER in ASR and ATWV in KWS. It also arouse our curiosity that the result of MFCC is better than FBANK in CTC model.

5. Conclusion and Future Work

A keywords spotting system is constructed based on a speech recognition system whose acoustic model is trained with recurrent neural network using CTC. The weighted finite state transducers were constructed for the decoding lattice and the keywords, respectively. The keyword spotting is conducted on these two WFSTs. Experiments were carried out to evaluate the effectiveness of the proposed technique. An appreciate beam width and acoustic scale are investigated. When the model is trained on audio data of 250 hours from RASC863, the ATWV and MTWV are 0.8310 and 0.8328 respectively. The ATWV is improved by 6.32% relatively compared with traditional DNN-HMM. It is due to that CTC models the initials and finals of Mandarin syllables directly.

We plan to try to model other levels of phonetic elements, such as characters or syllables, to examine the appropriate elements for keywords spotting in Mandarin. The reason why the MFCC feature is more effective than FBANK feature is considered to investigate. We will also consider that the model can be trained for KWS task directly.

6. Acknowledgements

This work is supported by the National High-Tech Research and Development Program of China (863 Program) (No.2015AA016305).

7. References

[1] Silaghi, Marius Calin, and R. Vargiya. "A new evaluation criteria for keyword spotting techniques and a new algorithm." in *INTERSPEECH,* 2005, pp.1593-1596.

[2] Mandal, Anupam, K. R. P. Kumar, and P. Mitra. "Recent developments in spoken term detection: a survey." *International Journal of Speech Technology* vol.17, no.2, pp.183-198, 2014.

[3] Zhang, Yaodong, and James R. Glass. "Unsupervised spoken keyword spotting via segmental DTW on Gaussian posteriorgrams." in *2009 Automatic Speech Recognition & Understanding (ASRU)* IEEE, 2009.

[4] J.R. Rohlicek, W. Russell, S. Roukos, and H. Gish, "Continuous hidden Markov modeling for speaker-independent wordspotting," in *Proceedings of the International Conference on Acoustics, Speech and Signal Processing (ICASSP).* IEEE*,* 1990, pp. 627–630.

[5] Chen, Guoguo, Carolina Parada, and Georg Heigold, "Small-footprint keyword spotting using deep neural networks." in *International Conference on Acoustics, Speech, and Signal Processing (ICASSP).* IEEE*,* 2014

[6] Chelba, Ciprian, and Alex Acero, "Position specific posterior lattices for indexing speech." in *Meeting of the Association for Computational Linguistics*, 2005.

[7] B.-H JUANG, "Maximum-Likelihood Estimation for Mixture Multivariate Stochastic Observations of Markov Chains Maximum-Likelihood Estimation for Mixture." *IEEE Transactions on Information Theory* vol.32, no.2, pp.307-309, 2015.

[8] Hinton, G., et al. "Deep Neural Networks for Acoustic Modeling in Speech Recognition." *IEEE Signal Processing Magazine* vol.29, no.6, pp. 82-97, 2012 .

[9] Mohri, Mehryar, F. Pereira, and M. Riley. "Weighted finite-state transducers in speech recognition." *Computer Speech & Language,* vol.16, no.1, pp. 69-88,2002.

[10] Dixon, Paul R., et al. "Recent Development of WFST-Based Speech Recognition Decoder."

[11] Graves, A., and N. Jaitly. "Towards end-to-end speech recognition with recurrent neural networks." in *International Conference on Machine Learning*, pp. 1764-1772, 2014.

[12] Bahdanau, Dzmitry, et al. "End-to-End Attention-based Large Vocabulary Speech Recognition." *Computer Science*, 2015.

[13] Miao, Yajie, M. Gowayyed, and F. Metze. "EESEN: End-to-end speech recognition using deep RNN models and WFST-based decoding." in *Automatic Speech Recognition and Understanding (ASRU).* IEEE, 2015.

[14] Can, Dogan, and Murat Saraclar, "Lattice Indexing for Spoken Term Detection." *IEEE Transactions on Audio, Speech, and Language Processing*, vol19, no.8, pp: 2338-2347, 2011.

[15] Chen, Guoguo, et al. "Using proxies for OOV keywords in the keyword search task." in *Automatic Speech Recognition and Understanding (ASRU).* IEEE,2013

[16] Bisani, Maximilian, and H. Ney. "Joint-sequence models for grapheme-to-phoneme conversion." *Speech Communication* vol.50, no.5, pp. 434-451, 2008.

[17] D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, M. Hannemann, P. Motlicek, Y. Qian, P. Schwarz, J. Silovsky, G. Stemmer, and K. Vesely, "The Kaldi speech recognition toolkit," in *Automatic Speech Recognition and Understanding (ASRU).* IEEE, 2011.

[18] Werbos, Paul J., "Backpropagation through time: what it does and how to do it." *Proceedings of the IEEE*, vol78, no.10, pp: 1550-1560, 1990.

[19] Li A, Yin Z, Wang T, Fang Q, Hu F, "RASC863 - a Chinese speech corpus with four regional accents", in *ICSLT-o-COCOSDA, New Delhi, India,* 2004

[20] Stolcke, Andreas. "Srilm --- An Extensible Language Modeling Toolkit." in *International Conference on Spoken Language Processing*, pp. 901—904, 2015.

[21] —."NIST Open Keyword Search 2016 Evaluation," *Available at* http://nist.gov/itl/iad/mig/openkws16.cfm, 2016

[22] K. Veselý, et al. "Sequence-discriminative training of deep neural networks." in *INTERSPEECH*, 2013.