MULTI-STREAM COMBINATION FOR LVCSR AND KEYWORD SEARCH ON GPU-ACCELERATED PLATFORMS

Wonkyum Lee, Jungsuk Kim and Ian Lane

Carnegie Mellon University
Electrical and Computer Engineering
5000 Forbes Avenue, Pittsburgh, PA 15213-3890, USA
wonkyuml@cmu.edu, jungsuk@cmu.edu, lane@cs.cmu.edu

ABSTRACT

In this paper, we explore methods for system combination of acoustic models having different features, modeling approaches and phonetic decision trees for speech recognition and keyword search. We introduce a Graphic Processing Unit (GPU)-accelerated lattice generation method and show that this architecture is efficient and well suited for multi-stream acoustic model combination. Additionally, we introduce a novel method to combine acoustic models with different phonetic trees into a single fully composed HMM state level (H-level) WFST network allowing lattice generation to be performed using diverse acoustic models. We evaluate the performance of our multi-stream approach to three standard techniques and observe that multi-stream combination obtains higher speech recognition accuracy than Lattice Combination or ROVER (up to 5.5% relative improvement in speech recognition accuracy compared to the single best model). Additionally, at an equivalent runtime, multi-stream combination obtained a 15% higher Average Term Weighted Value (ATWV) compared to CombMNZ for the keyword search task. By combining phonetic decision tree, we obtained gain (WER reduction) from the diversity of phonetic decision tree by using more efficient tree for each acoustic model.

Index Terms— Multi-stream acoustic model combination, Keyword search, Weighted Finite State Transducer (WFST), Graphics Processing Units (GPU), OpenKWS 2013

1. INTRODUCTION

System combination is often applied to improve the accuracy of automatic speech recognition (ASR) and related tasks such as keyword search. By combining the output from multiple ASR systems, errors generated by an individual system can be mitigated, improving speech recognition accuracy. Common approaches for ASR system combination include; Recognizer Output Voting Error Reduction (ROVER) [1], which combines the 1-best hypothesis from each system, Confusion Network Combination (CNC) [2, 3] and Lattice Combination [4] techniques, where sets of hypotheses are combined across multiple systems, and multi-stream combination [5], in which acoustic model likelihoods are combined at the HMM-state level during decoding. For tasks such as keyword search it has been observed that rather than using one of the combination methods above, performance can be improved further by independently performing keyword search across different ASR systems and then combining the keyword search output using standard approaches used in information retrieval, such as CombMNZ [6] or WCombMNZ [7]. Using system combination we have observed improvements in speech recognition accuracy of up to 11% relative and improvements in keyword search performance, Actual Term Weighted Value (ATWV), of up to 47% relative compared to our best single system for the Vietnamese LimitedLP task in OpenKWS 2013.

Although system combination generally improves accuracy, it comes at a high computational cost, as each additional model introduced significantly increases the computation required. For a *N*-way system combination the computation cost and runtime speed is approximately a factor of *N* larger than a single system. Additionally, in order to obtain the good performance it is most effective to perform combination of models that have similar performance but are as diverse as possible, models that use different features, different modeling approaches, such as Gaussian Mixture Model (GMM) or Deep Neural Network (DNN) acoustic models, or different phonetic decision trees generally obtain the best performance when combined.

While priors works have investigated numerous methods for ASR system combination [1, 2, 3, 4, 5], there has been limited investigation on what type of models to combine, and diversity in which components of the speech recognition models (features, acoustic modeling approach, or phonetic decision tree) obtain the best combined performance. Additionally, priors works do not consider the computation cost or runtime speed of such combination methods. In this paper we investigate these two areas. We compare the performance of different combination approaches and evaluate the performance of combining models that differ by feature, acoustic modeling technique and phonetic decision tree. In order to combine acoustic models with different phonetic trees during multi-stream speech recognition we introduce a novel method that composes a single fully composed HMM state level (H-level) WFST network using multiple phonetic trees. In this approach we generate a set of virtual HMM-states, where each virtual state maps to a weighted set of states per acoustic model, and using these virtual states to compose a single H-level WFST network that is applied during decoding. Additionally, we explore lattice generation on GPU-accelerated platforms, and demonstrate that this architecture is well suited for multi-stream speech recognition. By performing acoustic-model likelihood computation on the GPU and lattice generation on the

Supported in part by the Intelligence Advanced Research Projects Activity (IARPA) via Department of Defense U.S. Army Research Laboratory (DoD/ARL) contract number W911NF-12-C-0015. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purpose notwithstanding any copyright annotation thereon. Disclaimer: The views and conclusions annotated herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of IARPA, DoD/ARL, or the U.S. Government.

This work is supported in part by the Samsung GRO program

978-1-4799-2893-4/14/$31.00 ©2014 IEEE

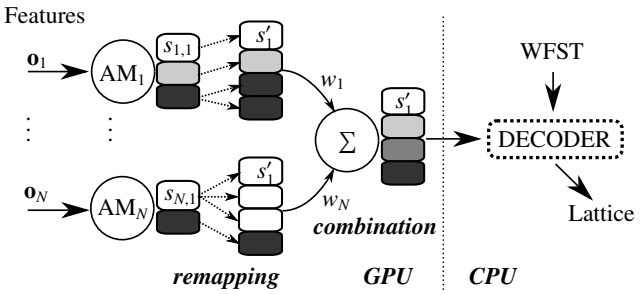

Fig. 1. Multi-stream acoustic model combination (w_i is combination weight for AM$_i$.)

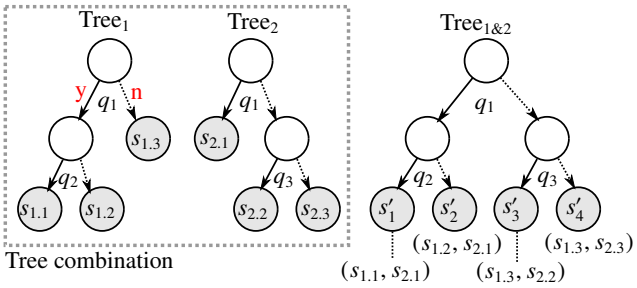

Fig. 2. Phonetic tree combination example.

CPU we are able to accelerate lattice generation by approximately 3x, and we see little degradation even when additional acoustic models are combined during decoding.

The rest of this paper is organized as follows. In Section 2, we review techniques to combine acoustic models. Section 3 describes the acoustic combination under WFST framework. Lattice generation on GPU-accelerated platform is presented in Section 4. Section 5 shows the experimental results. Finally, Section 6 concludes the paper and discusses future work.

2. TECHNIQUES FOR ACOUSTIC MODEL COMBINATION

In the speech to text (STT) task, system combination can be performed at the feature level, model level or recognition outputs. In STT task, the most popular method is the Recognizer Output Voting Error Reduction (ROVER) technique [1], which combines the 1-best results from multiple ASR systems into a composite word level network, which derives a single recognition hypothesis using majority voting. The two approaches of Confusion Network (CN) [2, 3] and Lattice Combination[4], both of which rely on Bayes decision rule to minimize word error rate (WER), were used to combine multiple hypothesis from each ASR system.

For keyword search, also known as spoken term detection, system combination is performed using CombMNZ[6] or WCombMNZ[7]. Although these techniques have been shown to improve ATWV, they require keyword search results from multiple systems and thus the computational cost increases linearly according to the number of ASR systems to be combined.

In addition to combination using lattices or 1-best output from a decoder it is also possible to combine acoustic model at the acoustic likelihood score. This method is generally known as a multi-stream acoustic model. When we combine acoustic score from individual acoustic model, several averaging methods can be used, including those listed below.

Arithmetic Mean Averaging:

$$\log(p_c(\mathbf{o}|s_k^{'})) = \frac{1}{N}\sum_{i=1}^{N} w_i \log(p_i(\mathbf{o}_i|s_{i,j})) \quad (1)$$

Geometric Mean Averaging:

$$\log(p_c(\mathbf{o}|s_k^{'})) = \sqrt[N]{\prod_{i=0}^{N} w_i \log(p_i(\mathbf{o}_i|s_{i,j}))} \quad (2)$$

Harmonic Mean Averaging:

$$\log(p_c(\mathbf{o}|s_k^{'})) = (\frac{1}{N}\sum_{i=0}^{N} w_i \log(p_i(\mathbf{o}_i|s_{i,j})^{-1}))^{-1} \quad (3)$$

where $\log(p_c(\mathbf{o}|s_k^{'}))$ is the combined log-likelihood score of the feature $\mathbf{o}$ given state $s_k^{'}$ in the combined decision tree, $\log(p_i(\mathbf{o}|s_{i,j}))$ is the log-likelihood score of the context dependent state $s_{i,j}$ at model i which is subject to map to state $s_k^{'}$ in combined tree, w_i is the weight parameter to re-scale log likelihood value according to the acoustic model and N is the number of acoustic models to be combined.

In the experimental evaluation we use Arithmetic mean as we found it to significantly outperform both Geometric Mean and Harmonic Mean for multi-stream combination.

3. ACOUSTIC MODEL COMBINATION UNDER WFST FRAMEWORK

Weighted finite state transducers (WFST) offer a unified framework for representing different knowledge sources and is well suited for speech recognition. In speech recognition, the phonetic, lexical and acoustic model can be composed together and optimized for speed and size ahead of decoding [8]. This enables decoder to be simpler and generally faster than dynamic decoders especially on the GPU-accelerated platforms [9].

Applying multi-stream acoustic models is not obvious under WFST frameworks. In order to take advantage of multiple acoustic models, knowledge from phonetic decision trees are required during search since each acoustic model could map a given context depend phone to different HMM states. In the dynamic search decoder, phonetic decision trees are available and can be used in order to find corresponding HMM states for different acoustic models. In standard WFST composition procedure, however, allows only a single phonetic decision tree per a HMM state level (H-level) WFST. In addition, mapping between context dependent phone to corresponding HMM is composed and encoded in H-level WFST network and hard to maintain during search. In the following section, we investigate two possible techniques in order to applying multi-stream acoustic models under WFST framework.

3.1. Combination with a common phonetic decision tree

One typical approach of applying multi-stream acoustic models in WFST based speech recognition is using a common phonetic decision tree for different acoustic models. For example, a DNN acoustic model trained over an alignment generated from a GMM acoustic model would have a same phonetic decision tree. In this case, a common H-level WFST can be composed with the common phonetic decision tree and then be used for decoding with {DNN,GMM} combined acoustic models by simply combining likelihoods as explained in Section 2. In many acoustic models, however, the optimal phonetic decision tree structure are different with feature types and acoustic model training schemes. Sharing common phonetic decision tree for different acoustic model could degrade speech recognition accuracy.

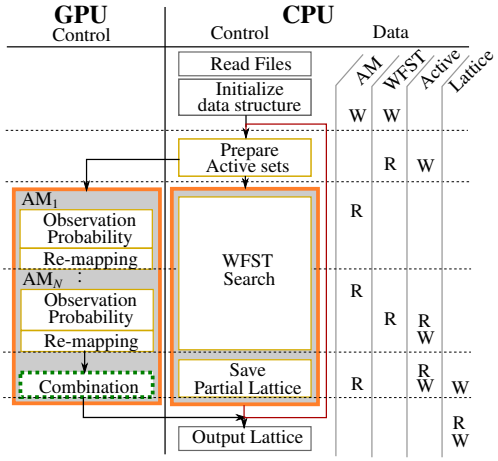

Fig. 3. Flowchart of the GPU-accelerated lattice generation with the multi-stream acoustic model.

3.2. Phonetic decision tree combination

To overcome the limitation of the WFST composition, we combined multiple phonetic decision trees into a single global tree which can then be composed as in standard WFST composition. The global tree maps a given context dependent phone to a virtual HMM state. Each virtual HMM state maps to a set of HMM states as defined in the original phonetic decision trees. Fig. 2 shows simple 2-way phonetic decision tree combination. For a given context dependent phone, tree_1 maps it to HMM state $s_{1.1}$ when answers of question set (q_1, q_2) are (y, y) while tree_2 maps it to $s_{2.1}$. In this scenario, the global tree $\text{tree}_{1\&2}$ maps it to a virtual HMM state s'_1 which maintain the mapping $s'_1 \rightarrow (s_{1.1}, s_{2.1})$. During decoding, log-likelihood score of the virtual HMM state s'_1 can be computed by combining the log-likelihood score of $s_{1.1}$ and $s_{2.1}$ based on this map.

4. LATTICE GENERATION ON GPU-ACCELERATED PLATFORM

While prior works [10, 11, 12] demonstrated the efficiency of using many-core GPU in speech recognition. They are not suitable for lattice generation and keyword search. Generating the lattice is usually slower than the standard Viterbi decoding since the decoder must keep all sub-optimal forward links at each frame rather than only the best backward link required to find the best path. Additionally, on GPU-accelerated platforms, these sub-optimal forward links must be copied from the GPU to the CPU for each time frame. This communication overhead makes the lattice generation inefficient on GPU-accelerated platform. For keyword search, large lattices are preferred as they contain more hypotheses to search.

In this work, observation likelihoods in Phase 1 is computed on the GPU, and Phase 2, graph traversal is performed on the CPU as explained in [13]. The speed-up obtained with this implementation is limited due to the overhead of transferring the frame-level log-likelihood score between the CPU and the GPU. The size of log-likelihoods scores, however, is relatively smaller than that of the intermediate lattice per frame. Therefore, it is often more efficient to compute the log-likelihood scores on the GPU while generating lattice on the CPU. In addition, As we are using physically independent hardware, it is possible to execute Phase 1 and Phase 2 concurrently as shown in Fig. 3. By computing log likelihoods for the next frame, dependency between Phase 1. and Phase 2 can be removed. Furthermore, multiple log-likelihood score computation can be conducted concurrently by incorporating multi-stream or multi-GPU techniques [14]. In this paper we generate lattices following the approach described in [15].

5. EXPERIMENTAL EVALUATIONS

In this paper, we performed evaluation on Vietnamese dataset which was recently released as the IARPA BABEL Program Vietnamese language collection IARPA-babel107b-v0.7[16]. Each Limited language pack training data consists of 10 hours of conversational speech, and a separate 20 hours of test speech data, which is used for evaluation.

5.1. Baseline system performance

For the baseline systems, we trained 3 HMM/GMM systems and 3 hybrid DNN/HMM systems on top of 3 different bottleneck features (BNF) as described in [17]. The bottleneck features were extracted from stacked auto-encoders having a bottleneck layer with 42 units, which were trained to predict context dependent states. We used 3 different features to make the corresponding acoustic models complementary. Since Vietnamese is tonal language, fundamental frequency variation feature(FFV feature) [18] and pitch tracking[19] were appended to the MFCC and log mel filterbank coefficients. Each BNF is described in Table 1.

Bottle-neck feature for Experiments			
BNF	Dimension	Input feature	Input frames
BNF1	42	lmel + FFV	11
BNF2	42	lmel + FFV + Pitch	11
BNF3	42	MFCC +FFV	11

Table 1. BNF feature for Training DNN and GMM models

All of GMM and DNN systems which are listed in Table 2 share the decision tree for the 2172 context dependent HMM states. GMM systems were trained with boosted maximum mutual information(BMMI) training. During maximum likelihood(ML) training stage, the alignment was extracted for DNN training. We trained DNN systems with 2 sigmoid hidden layers containing 2500 units each and soft-max output layer.

Model	AM	Feature	WER	ATWV
GMM1	GMM	BNF1	68.0%	0.1341
GMM2		BNF2	69.5%	0.1271
GMM3		BNF3	71.5%	0.1171
DNN1	DNN	BNF1	67.3%	0.1377
DNN2		BNF2	68.3%	0.1328
DNN3		BNF3	69.8%	0.1034

Table 2. Baseline systems in Vietnamese

The baseline results of Vietnamese are listed in Table 2. DNN1 shows the best performance in terms of WER(67.3%) and ATWV(0.1377). We took DNN1 as a baseline to see how much combining models gives gain over the different models and features. The same fixed beam was used for a fair comparison to produce similar size of lattices, which affect ATWV[20].

5.2. Combination for speech recognition accuracy

AM	Multi-stream	Lattice combination	Rover
	WER	WER	WER
1 Model (DNN1)	67.3%	67.3%	67.3%
2 Models	64.7% (-3.9)	66.1% (-1.8)	66.6% (-1.0)
4 Models	63.6% (-5.5)	64.1% (-4.8)	65.2% (-3.1)
6 Models	63.6% (-5.5)	63.8% (-5.2)	64.9% (-3.6)

Table 3. Combination for speech recognition accuracy on Vietnamese

Table 3 shows the comparison result of speech recognition accuracy between three different acoustic model combination approaches - multi-stream, lattice combination and ROVER. When we combine 4 acoustic models in multi-stream, we got error reduction by 5.5% relative while other lattice combination and ROVER got 4.8 % and 3.1% relative error reduction, respectively. Since Lattice Combination and ROVER require individual decoding result, their total run-time increases proportional to number of acoustic models being combined. However, multi-stream combination performs single pass decoding and therefore its run-time does not increase much even though 6 models are combined while its speech recognition accuracy outperforms the those of other methods.

5.3. Combination for keyword search

AM	Multi-stream	CombMNZ
	ATWV (RTF)	ATWV (RTF)
1 Model (DNN1)	0.1377 (0.62)	0.1377 (0.62)
2 Models	0.1663 (0.62)	0.1548 (1.24)
4 Models	0.1776 (0.63)	0.1890 (2.48)
6 Models	0.1794 (0.64)	0.2024 (3.72)
6 Models (large lattice)	0.2281 (3.60)	

Table 4. Combination for Keyword search on Vietnamese

Table 4 shows the comparison result of keyword search performance between three different acoustic model combination approaches - multi-stream and CombMNZ. When we combine 6 DNN models, we observed improvement in ATWV by 30% and 47% relative gain in multi-stream combination and CombMNZ, respectively. When we increase beam size(lattice size), where we got run-time similar to that of CombMNZ, multi-stream combination obtained a 15% higher ATWV than CombMNZ as shown in Table4. We can say that multi-stream combination on GPU-accelerated platform is the best combination method in keyword search task when an equivalent run-time is given.

5.4. Analysis of complementary features, models and phonetic decision trees

AM	WER	ATWV
DNN1	67.3%	0.1377
DNN1 + DNN2	65.9% (-2.1)	0.1577 (+14.5)
DNN1 + GMM1	65.3% (-3.0)	0.1628 (+18.2)
DNN1 + GMM2	64.7% (-3.9)	0.1663 (+20.8)

Table 5. Performance of multi-stream combination with different model structures and features in Vietnamese

Here, we want to point out that which diversity of models and features helps multi-stream combination more. From the Table 5, we see that when we combine DNN and GMM models with same features (DNN1+GMM1), the gain was larger than when we combine DNN models with different features (DNN1+DNN2) in terms of both WER and ATWV. It says that different model structures such as GMM and DNN models are more important than different features. By combining GMM and DNN models that also differ in feature set (DNN1+GMM2), additional improvement was obtained over the different models with same feature (DNN1+GMM1).

Model	Phonetic Decision Tree	Feature	WER	ATWV
DNN1	w/o Tree Diversity	BNF1	67.3%	0.1377
DNN2		BNF2	68.3%	0.1328
DNN3		BNF3	69.8%	0.1034
DNN1	w/ Tree Diversity	BNF1	67.3%	0.1377
DNN4		BNF2	67.7%	0.1424
DNN5		BNF3	69.2%	0.1201

Table 6. DNN systems with different phonetic decision tree

The proposed approach for combining phonetic decision enabled us to build acoustic model without any restriction on tree, we were able to build better individual acoustic models by using more efficient tree for each corresponding acoustic feature and model. Here, we rebuilt DNN4 and DNN5 systems by using different phonetic decision trees which seemed to be more efficient to deal with BNF2 and BNF3, respectively. As shown in Table 6, DNN4 and DNN5 performs better than DNN2 and DNN3, which were trained on same tree that DNN1 have been using.

AM	WER	ATWV
3 DNNs Combination(w/o Tree Diversity)	65.8%	0.1520
3 DNNs Combination(w/ Tree Diversity)	65.3%	0.1493

Table 7. Performance of multi-stream combination with different phonetic decision tree

As shown in Table 7, combination of DNN acoustic models having different phonetic decision tree performs better in terms of WER while there was no improvement on ATWV. The diversity of phonetic decision tree is shown to extend the room for optimizing individual acoustic models to be combined in multi-stream combination. By this, we were able to obtain additional gain in multi-stream combination.

6. CONCLUSION

In this paper we investigated the methods to combine multiple acoustic models having different features, model structures and phonetic decision tree for LVCSR and keyword search task. By proposing GPU-accelerated method to combine acoustic models and to generate lattice, multi-stream acoustic model combination was efficiently performed in GPU platform. Our experiments in Vietnamese task, multi-stream combination performs better than Lattice Combination or ROVER in speech recognition accuracy with 5.5% error reduction from single best model. Also it was shown that multi-stream acoustic model combination outperform CombMNZ when an equivalent run-time is given for keyword search task. Our proposed approach to combine phonetic decision tree enabled us to obtain gain from the diversity of phonetic decision tree by using more efficient tree for each acoustic model.

7. REFERENCES

[1] J. G. Fiscus, "A post-processing system to yield reduced word error rates: Recognizer output voting error reduction (rover)," in *ASRU*. IEEE, 1997.

[2] L. Mangu, E. Brill, and A. Stolcke, "Finding consensus in speech recognition: word error minimization and other applications of confusion networks," *Computer Speech & Language*, vol. 14, no. 4, pp. 373–400, 2000.

[3] G. Evermann and P.C. Woodland, "Posterior probability decoding, confidence estimation and system combination," in *NIST Speech Transcription Workshop*, 2000.

[4] F. Wessel, R. Schluter, and H. Ney, "Explicit word error minimization using word hypothesis posterior probabilities," in *ICASSP*. IEEE, 2001.

[5] H. Hermansky, "Multistream recognition of speech: Dealing with unknown unknowns," *Proceedings of the IEEE*, vol. 101, no. 5, pp. 1076–1088, 2013.

[6] Edward A Fox and Joseph A Shaw, "Combination of multiple searches," *NIST SPECIAL PUBLICATION SP*, pp. 243–243, 1994.

[7] Jonathan Mamou, Jia Cui, Xiaodong Cui, Mark JF Gales, Brian Kingsbury, Kate Knill, Lidia Mangu, David Nolden, Michael Picheny, Bhuvana Ramabhadran, et al., "System combination and score normalization for spoken term detection," in *ICASSP*, 2013.

[8] M. Mohri, F. Pereira, and M. Riley, "Weighted Finite-State Transducers in Speech Recognition," *Computer Speech and Language*, vol. 16, no. 1, pp. 69–88, 2002.

[9] J. Chong, E. Gonina, K. You, and K. Keutzer, "Exploring Recognition Network Representations for Efficient Speech Inference on Highly Parallel Platforms," in *Proc. Interspeech*, Sep. 2010, pp. 1489–1492.

[10] J. Chong, E. Gonina, Y. Yi, and K. Keutzer, "A Fully Data Parallel WFST-based Large Vocabulary Continuous Speech Recognition on a Graphics Processing Unit," in *Proc. Interspeech*, Sep. 2009, pp. 1183–1186.

[11] J. Kim, K. You, and W. Sung, "H- and C-level WFST-based Large Vocabulary Continuous Speech Recognition on Graphics Processing Units," in *Proc. ICASSP*, May 2011, pp. 1733–1736.

[12] K. You, J. Chong, Y. Yi, E. Gonina, C. J. Hughes, Y.-K. Chen, W. Sung, and K. Keutzer, "Parallel Scalability in Speech Recognition," *IEEE Signal Processing Magazine*, vol. 26, no. 6, pp. 124–135, Nov. 2009.

[13] P. R. Dixon, T. Oonishi, and S. Furui, "Fast Acoustic Computations using Graphics Processors," in *Proc. ICASSP*, Apr. 2009.

[14] NVIDIA, *NVIDIA CUDA Programming Guide Version 5.0*, May 2012.

[15] D. Povey, M. Hannemann, G. Boulianne, L. Burget, A. Ghoshal, M. Janda, M. Karafiát, S. Kombrink, P. Motlícek, Y. Qian, K. Riedhammer, K. Veselý, and N. T. Vu, "Generating exact lattices in the wfst framework," in *Proc. ICASSP*, 2012, pp. 4213–4216.

[16] IARPA, "Iarpa babel program," http://www.iarpa.gov/Programs/ia/Babel/babel.html.

[17] J. Gehring, W. Lee, K. Kilgour, I. Lane, Y. Miao, and A. Waibel, "Modular combination of deep neural networks for acoustic modeling," in *INTERSPEECH*, 2013.

[18] K. Laskowski and Q. Jin, "Modeling instantaneous intonation for speaker identification using the fundamental frequency variation spectrum," in *ICASSP*. IEEE, 2009.

[19] K. Schubert, *Pitch tracking and his application on speech recognition*, Diploma Thesis at University of Kalsruhe(TH), 1998.

[20] Jia Cui, Xiaodong Cui, J Mamou, B Kingsbury, B Ramabhadran, L Mangu, M Picheny, A Sethy, and J Kim, "Developing speech recognition systems for corpus indexing under the iarpa babel program," in *ICASSP*, 2013.