PRONUNCIATION AND SILENCE PROBABILITY MODELING FOR ASR

Guoguo Chen[1], *Hainan Xu*[1], *Minhua Wu*[1], *Daniel Povey*[1,2], *Sanjeev Khudanpur*[1,2]

[1]Center for Language and Speech Processing
[2]Human Language Technology Center of Excellence
The Johns Hopkins University, Baltimore, MD 21218, USA
guoguo@jhu.edu, hxu31@jhu.edu, mwu56@jhu.edu, dpovey@gmail.edu, khudanpur@jhu.edu

Abstract

In this paper we evaluate the WER improvement from modeling pronunciation probabilities and word-specific silence probabilities in speech recognition. We do this in the context of Finite State Transducer (FST)-based decoding, where pronunciation and silence probabilities are encoded in the lexicon (L) transducer. We describe a novel way to model word-dependent silence probabilities, where in addition to modeling the probability of silence following each individual word, we also model the probability of each word appearing after silence. All of these probabilities are estimated from aligned training data, with suitable smoothing. We conduct our experiments on four commonly used automatic speech recognition datasets, namely Wall Street Journal, Switchboard, TED-LIUM, and Librispeech. The improvement from modeling pronunciation and silence probabilities is small but fairly consistent across datasets.

Index Terms: automatic speech recognition, pronunciation probability, silence probability

1. Introduction

Pronunciation modeling is an important task for automatic speech recognition (ASR), which bridges the gap between the baseform pronunciations defined in the lexicon and the actual pronunciations of words uttered in speech. Pronunciations can be modeled either explicitly or implicitly [1].

Explict pronunciation modeling usually views the pronunciation of a word as a sequence of symbols, for example, phones. In this type of modeling, alternative pronunciations are often created and augmented to the existing lexicon to accommodate pronunciation variations. A big chunk of work has been focusing on discovery of the alternative pronunciations. In [2], the authors decode the training corpus with an automatic phone recognizer to obtain frequent alternative pronunciations of frequent words. In [3] and [4], phonological rules are used to generate alternative pronunciations of given words, while in [5] and [6], statistical decision trees are used instead. Other work lies in the estimation of the probabilities of alternative pronunciations. In [7] and [8] pronunciation probabilities are estimated according to their relative frequency in the training data, while in [9] pronunciation probabilities depend on dynamic features such as speaking rate, segment durations and pitch.

Implicit pronunciation modeling makes use of the underlying statistic model to account for pronunciation variations, and therefore removes the necessity of modifying the lexicon in theory. One reasonable way of modeling pronunciation changes implicitly is to view it as phoneme substitutions and share parameters (e.g., Gaussian densities) across phonetic models [10, 11, 1]. In such method parameters of baseform phonemes are tied with their alternative realizations, thus modeling alternative pronunciations. Others view pronunciations as a set of features, and as a result pronunciation variation is viewed as feature asynchrony and changes [12, 13].

While pronunciation modeling has been well covered in the literature, the modeling of silence between words, or more precisely pronunciations, has not caught a lot of attention. In [8] and [1], the authors propose to append three different silence types to each lexicon entry: a non-silence type, a short pause and a long silence. They show in their experiments that the modeling of inter-pronunciation silence together with pronunciation probabilities gives obivous improvement in their tasks. Their silence model, however, is shared across all pronunciations. This is not optimal since certain words, or more precisely pronunciations, are more likely to have silence preceding or following them.

We believe it is beneficial to have a pronunciation dependent silence model, which models how likely silence is around any given pronunciation. In this paper, we propose to model pronunciation and silence jointly. We start off by estimating pronunciation probabilities for each lexicon entry from training data alignment, same as that in [8] and [1]. But instead of building global silence models for each pronunciation entry, we move further to build a joint model for pronunciations and silence. We conduct experiments on various speech recognition datasets and show that by estimating pronunciation probabilities, an averaged 0.8% relative word error rate reduction can be achieved, while the reduction given by the joint model is 2.3%.

The remainder of this paper is organized as follows. We briefly describe the generation of training data alignment in Section 2. We then explain how we estimate pronunciation probability in Section 3 and the joint pronunciation and silence model in Section **??**. We describe how we encode pronunciation probability and the joint pronunciation and silence model to the lexicon finite state transducer (FST) in Section 5. Experiment setup is detailed in Section 6, and results are shown in Section 7. Finally in Section 8 we reiterate our main claims.

2. Training Data Alignment

3. Pronunciation Probability Estimation

Lexicons used by modern LVCSR systems typically map words to one or more hand-written baseform pronunciations. They contain multiple pronunciations for certain words to accommodate pronunciation variations in continuous speech. Ideally, if a word has more than one baseform pronunciations, probabilities should also be associated with them to indicate how likely each pronunciation is, as one baseform pronunciation is likely to be uttered more frequently than another in given domain. However, most commonly available lexicons, for example CMUdict [14], does not provide such prior knowledge.

In this section, we propose to learn pronunciation probabilities from training data. We assume a lexicon that contains multiple pronunciations for certain words is available, and our task is to estimate pronunciation probabilities for those words. The discovery of alternative pronunciations, which has been done in previous work such as [5], is out of scope of this paper.

We start off by assigning probability 1 to each pronunciation entry in the given lexicon. Different baseform pronunciations of the same word all have probability 1 because we apply the "max-normalization" technique described in [15], where the probability of the pronunciation is normalized by the probability of the best pronunciation given the word. We then use this lexicon to train a triphone system, with which we force-align all the training data, and work out "word pronunciation" pairs. Let p_i be the i^{th} pronunciation of word w, where $1 \leq i \leq N_w$ and N_w is the number of different baseform pronunciations of word w in the given lexicon. Let $C(w, p_i)$ be the count of "w p_i" pairs in the training data, probability of pronunciation p_i given word w is computed as follows

$$P(p_i|w) = \frac{C(w, p_i) + \lambda_1}{\sum_{i=1}^{N_w}(C(w, p_i) + \lambda_1)} \tag{1}$$

where λ_1 is a smoothing term that we typically set to 1. We again apply the "max-normalization" technique and compute the normalized probability as follows

$$P'(p_i|w) = \frac{P(p_i|w)}{\max_{1 \leq i \leq N_w} P(p_i|w)} \tag{2}$$

The above quantity $P'(p_i|w)$ is in fact not a well defined probability any more due to the normalization. It will later be encoded into the lexicon FST to reflect how likely each pronunciation is. Details will be described in Section 5.

4. Silence Probability Estimation

This section explains how we model the probability of silence between words. We assume that silence is not modeled as part of the language model itself, since frequently language modeling data comes from sources that lack acoustic data and so cannot be aligned. In addition, since our goal is to model the silence probability as part of the lexicon transducer we limit ourselves to models that can be represented with minor modifications to the lexicon transucer (L).

We intend to allow optional silence at the beginning and end of every word sequence, and between each word in the sequence. Let us write the word sequence as $w_1 w_2 \ldots w_N$. In order to avoid having to treat the beginning and end of sentence as special cases, we treat the beginning-of-sentence symbol <s> and the end-of-sentence symbol </s> as part of this sequence, so w_1 is always <s> and w_N is always </s> . Then we can limit ourselves to computing the probability of silence between a pair of words. In addition, because the silence probability may in general depend on which pronunciation was used for the word, we treat each distinct pronunciation of each word as a separate word for purposes of silence-pronunciation probability estimation: that is, the notation w really means the word-pronunciation pair rather than the word itself, and in the rest of this section, when we say "word" we really mean word-pronunciation pair.

In the training data alignment, we consider that there is either a silence s or a non-silence n between each pair of words, so that a sequence $w_1\, w_2\, w_3\, w_4$ might be realized as something like $w_1\, s\, w_2\, n\, w_3\, s\, w_4$. As a generative model the framework will be that our language model (and pronunciation model) generate the sequence of words, and afterward we generate the s and n symbols.

4.1. Probability of silence to the right of a word

The first part of our model gives us the probability of silence s or non-silence n given the word immediately to the left. We use the notation $P(s_r|w)$ to mean the probability of silence appearing to the right of word w, and $P(n_r|w)$ to mean the probability of non-silence to the right of word w; these two sum to one. We compute $P(s_r|w)$ from training data counts using the following equation:

$$P(s_r|w) = \frac{C(w\, s) + \lambda_2 P(s)}{C(w) + \lambda_2}, \tag{3}$$

where $C(w\, s)$ is the count in the training data of the sequence $w\, s$, $C(w)$ is the count of word w in the training data, $P(s) = C(s)/(C(s) + C(n))$ is the overall probability of silence, and λ_2 is a smoothing constant that we set to 2 for experiments reported here. $P(s_r|w)$ is given as $1 - P(s_r|w)$.

4.2. Probability of silence to the left of a word

It is quite possible to compute the probability of silence (or non-silence) appearing to the left of a given word, i.e. as $P(s_l|w)$. In fact, analysis of the data counts as well as WERs (not included in this paper) leads us to believe that the word to the right (of the silence or non-silence) is a slightly better predictor than the word to the left. These experiments were done on conversation English speech and may not be equally true for all domains. However, ideally we would like to use both sources of information.

We sought a solution that could be efficiently encoded in L; this rules out using counts of triples such as $C(w_1\, s\, w_2)$. We wanted something more effective than simple score interpolation, and something which would avoid the problem of double-counting. By the problem of double-counting, what we mean is that if there is a pair of words that always appear in sequence, such as "san francisco", we needed a model that would give a reasonable probability of the silence in between them.

Our solution is to include a factor in the silence probabilities that is based on the word immediately to the right, but to estimate this as a "correction term" rather than as a probability. The aim is to capture whatever effect is not already modeled by the word to the left. We compute the correction terms, with smoothing, as follows:

$$F(s_l|w) = \frac{C(s\, w) + \lambda_3}{\bar{C}(s\, w) + \lambda_3} \tag{4}$$

$$F(n_l|w) = \frac{C(n\, w) + \lambda_3}{\bar{C}(n\, w) + \lambda_3} \tag{5}$$

where $\bar{C}(sw)$ and $\bar{C}(nw)$ are "fake counts" generated from our silence model considering only the word to the left of silence:

$$\bar{C}(sw) = \sum_v C(v\, ?\, w) P(s_r|v), \tag{6}$$

where the summation $\sum_v$ is taken over all words; the expression $C(v\, ?\, w)$ means the count of v, then silence or nonsilence, then w; and $P(s_r|v)$ is as computed in Equation (3). λ_3 is a smoothing term that we take to be 2 for experiments reported here.

4.3. Putting it all together

Considering the space between two words w_1 to the left and w_2 to the right, then the probability our model gives to silence appearing there is $P(s_r|w_1)F(s_l|w_2)$, where $P(s_r|w_1)$ is computed as in Equation (3) and $F(s_l|w_2)$ is computed as in Equation (**??**); and the probability of nonsilence is $P(n_r|w_1)F(n_l|w_2)$ where $F(n_l|w_2)$ is computed as in Equation (**??**). These two quantities may not sum exactly to one; this is the price we pay for having a model that factors this easily.

5. Lexicon Finite-State Transducer

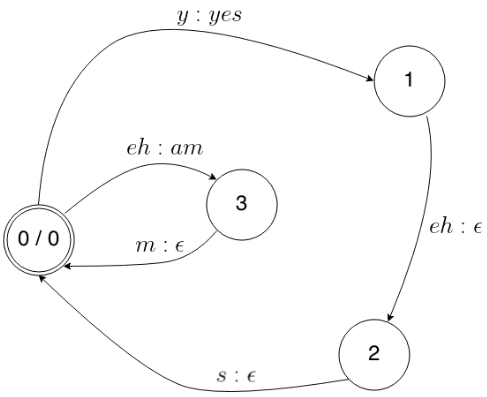


Figure 1: Basic lexicon WFST (L_1)

Speech recognition in weighted finite-state transducer (WFST) framework represents lexicon as a WFST, which maps a sequence of pronunciation phones to a sequence of words [16]. An example of a simple version of the lexicon WFST is depicted in Figure 1 (L_1), where only two vocabulary words *yes* → [y eh s] and *am* → [eh m] are allowed. L_1 does not model any pronunciation probability, and no silence is allowed between words.

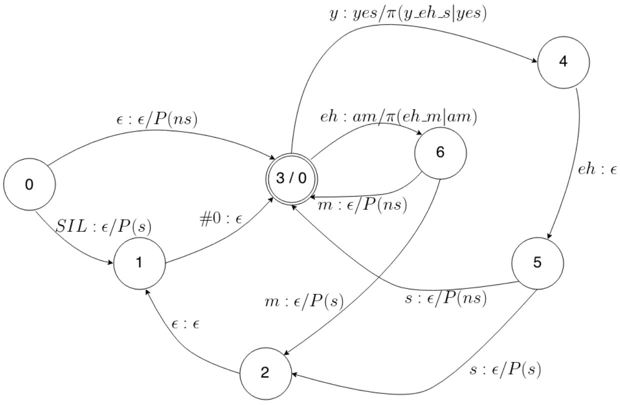
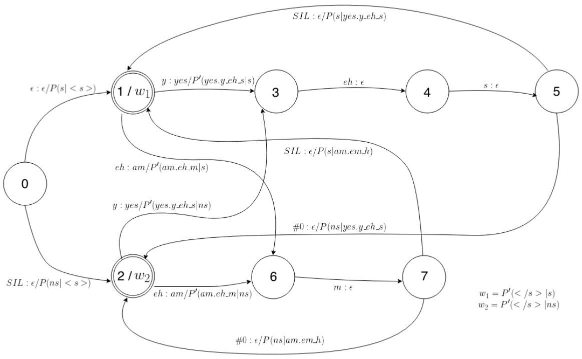


Figure 2: Lexicon WFST with optional silence and pronunciation probabilities (L_2)

A more reasonable way is to allow optional silence between words, since pause can occur around them in continuous speech. One caveat of using optional silence between words is that it will make the lexicon WFST non-determinizable, so disambiguation symbols should be added properly [16]. One may also add pronunciation probabilities (Section 3) as arc weights in the lexicon WFST, to make certain pronunciations more likely than others. Figure 2 illustrates the lexicon WFST with optional silence and pronunciation probabilities, which is used by the open source speech recognition toolkit Kaldi [17]. L_2 has three more states than L_1: a start state 0, representing the beginning of the sentence, a silence state 2 which inserts silence between words and a disambiguation state which adds disambiguation symbols after inserting the silence. Pronunciation probabilities are associated with each pronunciation and global silence probability can also be encoded when inserting silence.



$SIL : \epsilon/P(s|yes.y\_eh\_s)$
$\epsilon : \epsilon/P(s|<s>)$
1 / w_1
$y : yes/P'(yes.y\_eh\_s|s)$
3
$eh : \epsilon$
4
$s : \epsilon$
5
$SIL : \epsilon/P(s|am.em\_h)$
$eh : am/P'(am.eh\_m|s)$
0
$y : yes/P'(yes.y\_eh\_s|ns)$
$\#0 : \epsilon/P(ns|yes.y\_eh\_s)$
$SIL : \epsilon/P(ns|<s>)$
2 / w_2
6
7
$eh : am/P'(am.eh\_m|ns)$
$m : \epsilon$
$w_1 = P'(</s>|s)$
$w_2 = P'(</s>|ns)$
$\#0 : \epsilon/P(ns|am.em\_h)$

Figure 3: Lexicon WFST with joint pronunciation and inter-word silence model (L_3) the font seems to be too small – I'll change after we fix the notations.

To incorporate the joint pronunciation and inter-word silence model into the lexicon WFST, we have to use one more state called non-silence state, in addition to the silence state. An example WFST contains two vocabulary words is shown in Figure 3. Generally, if the current state is the beginning of sentence, or end of some words, then the WFST has to transit to either silence state or non-silence state first, with the corresponding probabilities estimated in Section 4. It then has to transit back to another word, or end of sentence, also with the probabilities described in Section 4. Different from L_2, we remove the disambiguation state from L_3. We instead insert disambiguation symbols whenever we transit back to the non-silence state.

6. Experiment Setup

This section describes how we conduct our experiments. We use the open source speech recognition toolkit Kaldi [17] for all our implementation and experiments.

6.1. Baseline System

We use Kaldi's online version of multi-splice neural network recipe [18] as our baselines. We start off by training a GMM system, where 13 dimensional Mel-frequency cepstral coefficient (MFCC) [19] features are extracted, followed by a typical maximum likelihood acoustic training recipe that begins with a flat-start initialization of context-independent phonetic HMMs, and ends with speaker adaptive training (SAT). This GMM system is used to force-align training data and generate labels for neural network training.

We then extract 40 dimensional MFCC features to train an i-vector extractor. We train the i-vector extractor on mean normalized MFCC features, but extract them from MFCC features without mean normalization. This way the i-vector can capture the mean shifts along input dimensions, which will be available to the neural network so that it can learn necessary normalizations. The i-vector extractor computes 100 dimensional i-vector for each speaker, which will then be appended to the 40 dimensional MFCC features to form features for neural network training.

Combining the training labels and features generated from the above two steps, we create examples for neural network, and perform multi-splice training. In the multi-splice architecture, splicing over time does not only happen at the input layer of neural network, but also at other layers such as the hidden layers. Besides, the splicing is usually over nonadjacent frames. For more details readers are referred to [18].

6.2. Datasets

Since previous work on pronunciation modeling usually yields modest gains [10], we decide to run our experiments over multiple datasets, to minimize the affect of noise. We conduct our experiments on four commonly used speech recognition datasets, namely the Wall Street Journal (WSJ) corpus [20], the Switchboard (SWBD) corpus [21], the TED-LIUM (TL) corpus [22] and the Librispeech (LS) corpus [23]. Note that for Librispeech only the 100 hour subset is used in our experiments.

dataset	pron / word ratio	% of multi-pron words
WSJ	1.077	7.2%
SWBD	1.024	1.7%
TL	1.051	4.5%
LS	1.033	3.0%

Table 1: Lexicon statistics for different datasets

We use the default lexicon in Kaldi recipes for each dataset. The pronunciation statistics of the lexicons are shown in Table 1. From the above table we can see that all lexicons have words with more than one pronunciations, especially for the WSJ lexicon, which contains more than 7% words with multiple pronunciations. All the lexicons come without prior knowledge for each pronunciation, which leaves room for improvement.

6.3. Analysis of using pronunciation model and joint model

		baseline	pron model	joint model
No	swbd	20.1%	20.0%	19.6%
	eval2000	26.9%	26.9%	26.7%
Yes	swbd	20.1%	19.9%	19.5%
	eval2000	26.9%	27.0%	26.7%

Table 2: WER of using (Yes) or not using (No) pronunciation model or joint model in training (SWBD SAT system)

The pronunciation model and joint model described in Section 3 and Section 4 are typically used for decoding, as done in [24]. They of course can also be used in training, which may improve the alignment and thus reduce the word error rate (WER). Before we run into the full experiments, we first conduct an analysis experiment on SWBD SAT system. Results in Table 2 suggest that using or not using those models in training will not make much a difference. Therefore, in the rest of our experiments, we only use the pronunciation model and joint model for decoding.

7. Results

		baseline	pron model	joint model
WSJ	eval92	4.1%	4.0%	3.9%
SWBD	swbd	13.7%	13.6%	13.1%
	eval2000	20.5%	20.4%	20.0%
TL	test	18.1%	17.9%	17.9%
LS	test_clean	6.7%	6.7%	6.7%
	test_other	24.0%	24.0%	23.7%

Table 3: WER performance of the pronunciation model and the joint model (DNN system) We're re-running the librispeech experiments, since the multisplice script we used was not tuned and was 2% worse than the one published in the paper

		pron model	joint model
WSJ	eval92	2.4%	4.9%
SWBD	swbd	0.7%	4.4%
	eval2000	0.5%	2.4%
TL	test	1.1%	1.1%
LS	test_clean	0%	0%
	test_other	0%	1.25%
average	-	0.8%	2.3%

Table 4: Relative WER reduction of the pronunciation model and the joint model when compared with baseline (DNN system)

Table 3 shows the WER performance of using pronunciation model and the joint model, and Table 4 gives the corresponding relative WER reduction when compared with the baseline. First, let us compare the numbers in Table 1 and Table 4. We can see from the two tables that generally if there are more words with multiple pronunciations, there will be more WER reduction when we model pronunciation probabilities. The reduction of from the pronunciation model alone, however, is modest (e.g., 0.8% averaged relative WER reduction), which is consistent with the discussion in previous [10]. Our joint pronunciation and inter-word silence model, on the other hand, gives an averaged relative WER reduction of 2.3%, which is more than doubled of that from pronunciation model alone.

8. Conclusion

We have presented a novel way to jointly model pronunciation and inter-word silence. Our experiments over four commonly used speech recognition datasets suggest that while the modeling of pronunciation alone can only lead to modest recognition improvement, the joint modeling with inter-word silence gives respectable reduction in terms of word error rate.

9. Acknowledgement

This work was partially supported by NSF Grant No IIS 0963898 and an unrestricted gift from Google, Inc. (Award 2012-R2-106).

10. References

[1] T. Hain, "Implicit modelling of pronunciation variation in automatic speech recognition," *Speech Communication*, vol. 46, no. 2, pp. 171–188, 2005.

[2] T. Slobada and A. Waibel, "Dictionary learning for spontaneous speech recognition," in *Proceedings of Fourth International Conference on Spoken Language Processing (ICSLP)*, vol. 4. IEEE, 1996, pp. 2328–2331.

[3] E. Giachin, A. Rosenberg, and C.-H. Lee, "Word juncture modeling using phonological rules for HMM-based continuous speech recognition," in *Proceedings of the International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 1990, pp. 737–740.

[4] G. Tajchman, E. Foster, and D. Jurafsky, "Building multiple pronunciation models for novel words using exploratory computational phonology," in *Proceedings of EUROSPEECH*, 1995.

[5] E. Fosler, M. Weintraub, S. Wegmann, Y.-H. Kao, S. Khudanpur, C. Galles, and M. Saraclar, "Automatic learning of word pronunciation from data," in *Proceedings of the International Conference on Spoken Language Processing*, 1996.

[6] B. Byrne, M. Finke, S. Khudanpur, J. McDonough, H. Nock, M. Riley, M. Saraclar, C. Wooters, and G. Zavaliagkos, "Pronunciation modelling for conversational speech recognition: A status report from WS97," in *Proceedings of the Automatic Speech Recognition & Understanding (ASRU) Workshop*. IEEE, 1997, pp. 26–33.

[7] B. Peskin, M. Newman, D. McAllaster, V. Nagesha, H. Richards, S. Wegmann, M. Hunt, and L. Gillick, "Improvements in recognition of conversational telephone speech," in *Proceedings of the International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, vol. 1. IEEE, 1999, pp. 53–56.

[8] T. Hain, P. Woodland, G. Evermann, and D. Povey, "The CU-HTK march 2000 Hub5e transcription system," in *Proceedings of Speech Transcription Workshop*, vol. 1, 2000.

[9] M. Finke and A. Waibel, "Speaking mode dependent pronunciation modeling in large vocabulary conversational speech recognition," in *Proceedings of EUROSPEECH*, 1997.

[10] M. Saraclar, H. Nock, and S. Khudanpur, "Pronunciation modeling by sharing Gaussian densities across phonetic models," *Computer Speech & Language*, vol. 14, no. 2, pp. 137–160, 2000.

[11] M. Saraclar and S. Khudanpur, "Pronunciation ambiguity vs. pronunciation variability in speech recognition," in *Proceedings of the International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, vol. 3. IEEE, 2000, pp. 1679–1682.

[12] M. Finke, J. Fritsch, D. Koll, and A. Waibel, "Modeling and efficient decoding of large vocabulary conversational speech." in *Proceedings of Eurospeech*, 1999.

[13] K. Livescu and J. Glass, "Feature-based pronunciation modeling for speech recognition," in *Proceedings of HLT-NAACL: Short Papers*. Association for Computational Linguistics, 2004, pp. 81–84.

[14] Cmudict. Visited 3/12/2015. [Online]. Available: https://http://www.speech.cs.cmu.edu/cgi-bin/cmudict

[15] J. Mamou and B. Ramabhadran, "Phonetic query expansion for spoken document retrieval," in *Proceedings of INTERSPEECH*. ISCA, 2008, pp. 2106–2109.

[16] M. Mohri, F. Pereira, and M. Riley, "Speech recognition with weighted finite-state transducers," in *Springer Handbook of Speech Processing*. Springer, 2008, pp. 559–584.

[17] D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, M. Hannemann, P. Motlíček, Y. Qian, P. Schwarz *et al.*, "The kaldi speech recognition toolkit," in *Proceedings of the Automatic Speech Recognition & Understanding (ASRU) Workshop*, 2011.

[18] V. Peddinti, D. Povey, and S. Khudanpur, "A time delay neural network architecture for efficient modeling of long temporal contexts," submitted to Interspeech 2015.

[19] S. Davis and P. Mermelstein, "Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences," *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. 28, no. 4, pp. 357–366, 1980.

[20] D. B. Paul and J. M. Baker, "The design for the Wall Street Journal-based CSR corpus," in *Proceedings of the workshop on Speech and Natural Language*. Association for Computational Linguistics, 1992, pp. 357–362.

[21] J. J. Godfrey, E. C. Holliman, and J. McDaniel, "SWITCHBOARD: Telephone speech corpus for research and development," in *Proceedings of the International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, vol. 1. IEEE, 1992, pp. 517–520.

[22] A. Rousseau, P. Deléglise, and Y. Estève, "TED-LIUM: an automatic speech recognition dedicated corpus." in *Proceedings of LREC*, 2012, pp. 125–129.

[23] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur, "Librispeech: an ASR corpus based on public domain audio books," in *Proceedings of the International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2015.

[24] V. Peddinti, G. Chen, D. Povey, and S. Khudanpur, "An i-vector based time delay neural network architecture for far field recognition." submitted to Interspeech 2015.