FRUSTRATINGLY EASY NOISE-AWARE TRAINING OF ACOUSTIC MODELS

Desh Raj[1], *Jesús Villalba*[1,2], *Daniel Povey*[3], *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.
[3]Xiaomi Corp., Beijing, China.

draj@cs.jhu.edu

ABSTRACT

Environmental noises and reverberation have a detrimental effect on the performance of automatic speech recognition (ASR) systems. Multi-condition training of neural network-based acoustic models is used to deal with this problem, but it requires many-folds data augmentation, resulting in increased training time. In this paper, we propose utterance-level noise vectors for noise-aware training of acoustic models in hybrid ASR. Our noise vectors are obtained by combining the means of speech frames and silence frames in the utterance, where the speech/silence labels may be obtained from a GMM-HMM model trained for ASR alignments, such that no extra computation is required beyond averaging of feature vectors. We show through experiments on AMI and Aurora-4 that this simple adaptation technique can result in 6-7% relative WER improvement. We implement several embedding-based adaptation baselines proposed in literature, and show that our method outperforms them on both the datasets. Finally, we extend our method to the online ASR setting by using frame-level maximum likelihood for the mean estimation.

Index Terms: robust speech recognition, noise-aware training, acoustic modeling, noise vectors

1. INTRODUCTION

Automatic speech recognition (ASR) systems are being increasingly deployed in real-life situations (e.g. home assistants like Amazon Echo, Google Home, etc.), where the acoustic environment may present distortions such as background noise and reverberation. To tackle these problems, the field of noise-robust ASR has organized various challenges over the past several years, such as the REVERB [1] or the CHiME [2, 3] challenges.

In this paper, we propose a new approach for noise-aware training of deep neural network (DNN) based acoustic models in hybrid ASR systems. We leverage a speech activity detection (SAD) model to identify the speech frames and the silence frames in the utterance. Our "noise vector", then, is obtained by concatenating the means of the speech frames and the silence frames, respectively. We conjecture that such a noise estimate is well-informed by the distortions occurring throughout the utterance, including additive and multiplicative noises. It is also more robust to insertion errors since the model is explicitly aware of the composition of non-speech frames. Furthermore, this additional information is acquired at no extra computational cost, since we make use of a GMM-HMM model that is already trained for generating alignments for training the DNN acoustic model.

In online speech recognition, we are only provided a partial input sequence, and the model is required to output streaming transcriptions. In such a setting, waiting until the whole utterance has been seen in order to compute the means of the speech and silence frames is not feasible, thus requiring online estimates of the noise vector. Towards this objective, we propose to use maximum likelihood (ML) estimates obtained from a probabilistic model of the means of speech and silence. This makes our noise-adaptive training scheme compatible with streaming ASR systems.

2. RELATED WORK

In the past, several methods have been proposed to make ASR systems robust to environmental noise [4], either through feature normalization (like spectral subtraction [5] and Weiner filtering [6]) or through model adaptation, such as MLLR-like schemes [7]. More recently, with the ubiquitous adoption of neural networks in hybrid as well as end-to-end ASR, data-driven supervised approaches have emerged as potential alternatives to traditional unsupervised methods [8]. These include front-end enhancement techniques like masking [9] or back-end methods like noise-aware and multi-condition training [10].

For robust streaming ASR, several online front-end processing methods have been proposed, such as time-frequency masks and steering vectors [11] or online-enabled Generalized Eigenvalue (GEV) beamformers [12]. In this work, we

This work was partially supported by grants from the JHU Applied Physics Laboratory, MIT Lincoln Labs, and the Government of Israel.

do not perform any front-end processing of the acoustic signals; instead, we obtain an informed estimate of the noise from the observed chunks of speech and non-speech frames, and use this estimate to adapt the acoustic model.

Several researchers have proposed the use of utterance-level embedding for noise/environment adaptation of acoustic models. Saon et al. [13] used i-vectors, proposed originally for speaker verification [14]. An "online" version of these i-vectors, estimated every 10 frames, is supported in the Kaldi [15] ASR toolkit. Seltzer et al. [10] used noise-aware training similar to our approach, but their noise vectors were estimated by averaging the first and last 10 frames in the utterance, whereas we estimate noise by leveraging a SAD module. In [16], the authors trained a network to predict the environment label, and used the embedding from the bottleneck-layer to inform the acoustic model. Similarly, [17] trained a linear discriminant analysis (LDA) model using environment labels to project i-vectors into a space that more accurately represents noise variability in the utterance, calling the resulting vectors as e-vectors. They also used a bottleneck DNN similar to [16] for estimating e-vectors, with the exception that they used i-vectors as input, instead of the acoustic features. We will look at these approaches in more detail in Section 4.2.

Since we use the means of acoustic features, comparisons may also be drawn to the popular cepstral mean normalization (CMN) [18] approach. However, while CMN just subtracts the cepstral mean, our method allows the DNN to learn non-linear transforms of the noise vector.

3. NOISE VECTORS

We make the acoustic model noise-aware by appending additional information with the input feature sequence in the form of "noise vectors." Let $\mathbf{x}_i = (\mathbf{x}_{i1}, \ldots, \mathbf{x}_{it}, \ldots, \mathbf{x}_{iT_i})$, $\mathbf{x}_{it} \in \mathbb{R}^d$, be the input feature frames corresponding to the utterance i. Consider an oracle SAD function, $f : \mathbb{R}^d \rightarrow \{\text{speech}, \text{sil}\}$, $f(\mathbf{x}_{it}) = s_{it}$, which classifies a frame as speech or silence. Let $\mathbf{s}_i = (s_{i1}, \ldots, s_{it}, \ldots, s_{iT_i})$ be the corresponding SAD predictions for the input sequence. We compute the means of speech and silence frames, $\boldsymbol{\mu}_{\text{s}_i}$ and $\boldsymbol{\mu}_{\text{n}_i}$, as

$$\boldsymbol{\mu}_{\text{s}_i} = \frac{\sum_{t \in \text{speech}} \mathbf{x}_{it}}{\sum_{t \in \text{speech}} 1}, \quad \text{and} \quad \boldsymbol{\mu}_{\text{n}_i} = \frac{\sum_{t \in \text{sil}} \mathbf{x}_{it}}{\sum_{t \in \text{sil}} 1}. \qquad (1)$$

In our hybrid ASR system, the acoustic model is trained using a sequence discriminative criterion [19] such as lattice-free MMI [20], on alignments generated using a GMM-HMM model with the training transcriptions. Since we know which phones map to silence phones, we use the same alignments to obtain the SAD label sequence $\mathbf{s}_i$. An overview of our noise-aware training scheme is shown in Fig. 1. We append the noise vectors to the input features through a linear transformation (called a "control layer" in literature [21])[1].

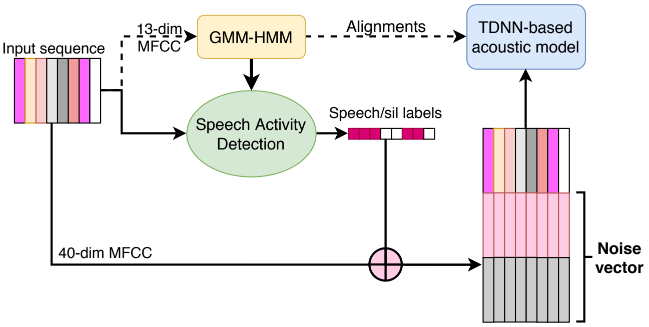

Fig. 1: Noise-aware training/inference pipeline with a TDNN-based acoustic model. Dotted lines denote paths that are only required during training.

At inference time, we run a first-pass decoding with the same GMM-HMM model to obtain the decoding lattice, which is then used to obtain the SAD labels for estimating the noise vectors. Since we only need approximate representations of speech and noise frames, the classification need not be perfect, and so this strategy suffices. Next, we describe an extension of this estimation technique to the online decoding scenario using maximum likelihood estimate.

Maximum likelihood (ML) estimate

The ML estimate of the noise vectors is similar to equation (1), with $\mathbf{s}_i$ being the partial sequence observed until time t. If no speech or silence frames are observed, the corresponding components are set to 0. In Section 5.3, we will show that this estimate of the noise vectors quickly converges to the offline estimate in most settings.

4. EXPERIMENTAL SETUP

4.1. Datasets and model

We performed experiments on two different datasets: AMI and Aurora-4. AMI [23] consists of 100 hours of recorded meetings, and speech from close-talk, single distant microphone (SDM), and array microphones is provided. We used the SDM recordings for our experiments, which is ideal for single-channel far-field evaluation. Aurora-4 [24] is an 80-hour close-talk speech corpus based on Wall Street Journal (WSJ) data with artificially added noise. We evaluated on the `eval92` and `0166` subsets. All our experiments were conducted using the Kaldi speech recognition toolkit [15] using neural networks trained with lattice-free MMI [20]. The acoustic model for Aurora-4 was based on factored TDNN [25], while the one for AMI was TDNN-based. The hyperparameters for each setup are shown in Table 1. For the Aurora-4 setup, we used a pruned 3-gram language

[1] See `fixed-affine-layer` in Kaldi recipes, and its resemblance to the LDA feature transformation [22].

Table 1: Hyperparameters for the acoustic model training for our experimental setups.

Setup	Model	# layers	Dim	Bottleneck	# epochs
AMI	TDNN	9	450	-	9
Aurora-4	TDNN-F	12	1024	128	10

model (LM) trained on WSJ transcriptions using the IRSTLM toolkit [26]. For AMI, we trained a pruned 3-gram LM with KN-smoothing on a combination of AMI and Fisher [27] data using the SRILM toolkit [28].

4.2. Baselines

As described earlier in Section 2, several embedding-based methods have been proposed for noise-aware training of acoustic models. Here, we perform comparisons with 4 of these methods, described below.

1. **i-vector**: i-vectors were first introduced in [14] for speaker verification, and later used for speaker adaptation of acoustic models in [13]. They are generative embeddings which model both speaker and channel variability in utterances. We used two variants in our experiments: *offline*, which were estimated over the whole utterance, and *online*, which were estimated every 10 frames, based on Kaldi recipes. For both these variants, we extracted 100-dim i-vectors using extractors trained on the corresponding training sets (for each dataset).

2. **NAT-vector**: Seltzer et al. [10] proposed "noise vectors" for noise-aware training, by averaging the first and last 10 frames of the utterance. We refer to these vectors as NAT-vectors, to avoid confusion with our proposed noise vectors. Since we use 40-dim MFCC features for training our models, these vectors are also 40-dimensional.

3. **e-vector**: Since i-vectors capture both speaker and channel variability, [17] proposed a technique for extracting the "environment" variability by transforming the i-vectors using environment labels, either using linear discriminant analysis (LDA), or bottleneck neural networks, and called them e-vectors. Here, we use the LDA variant of e-vectors for our comparison. Since Aurora-4 and AMI do not have annotated environment labels, we trained the LDA on CHiME-4 data [3], which consists of 4 different environments. We used this LDA to transform the 100-dim i-vectors to 50-dim e-vectors, following [17] where the authors did not see any WER improvement for e-vectors with higher dimensions.

4. **Bottleneck NN**: Since the above baselines estimate environment embeddings at the utterance level, Kim et al. [16] argued that they cannot effectively capture non-stationary noise. Instead, they trained a bottleneck DNN to classify noise types with frame-level acoustic features provided as input. Frame-level noise embeddings were then extracted from the bottleneck layer and used for noise-aware training. For our experiments, we trained a 5-layer feed-forward neural network with 1024 hidden units (similar to [16]), and a 80-dim bottleneck was placed at the 4th layer. Similar to the e-vector LDA training, we trained the bottleneck NN on the CHiME-4 data.

Table 2: WER results for our proposed noise vectors (offline and MLE) compared with other embedding-based techniques from literature. Methods marked with † require environment labels for training embedding extractor.

Method	AMI		Aurora-4	
	Dev	**Eval**	**Eval-92**	**0166**
Base model	38.8	42.6	7.94	8.12
CMN	38.5	42.8	7.79	7.94
utt-mean	36.8	40.8	7.77	7.78
i-vector (offline) [13]	35.2	39.4	8.24	8.53
i-vector (online)	35.6	40.0	8.51	8.78
NAT-vector [10]	38.1	42.0	8.17	8.38
e-vector (LDA)† [17]	37.6	41.5	7.69	7.91
Bottleneck NN† [16]	38.7	42.4	8.04	8.37
Noise vector (offline)	**34.9**	**38.8**	**7.37**	**7.61**
Noise vector (MLE)	35.4	39.2	7.72	7.93

Table 3: Insertion (ins), deletion (del), and substitution (sub) error comparison for noise vectors compared with the base model, for AMI (dev) and Aurora-4 (eval92).

Dataset	System	Ins	Del	Sub	WER
AMI (dev)	TDNN	4.7	12.6	21.5	38.8
	+ noise vector	3.2	11.5	20.3	34.9
Aurora-4 (eval92)	TDNN-F	0.50	2.57	4.87	7.94
	+ noise vector	0.58	2.12	4.67	7.37

In addition to these embedding methods from literature, we report results using (i) CMN, and (ii) embeddings obtained by averaging the entire utterance, which we call `utt-mean`. This ablation is performed to demonstrate the importance of obtaining separate means for speech and silence frames.

5. RESULTS AND DISCUSSION

5.1. Comparison with baselines

Table 2 shows the WER results for the proposed method and the baselines on AMI and Aurora-4. It can be seen from the table that adding offline noise vectors to the input features provided WER improvements of up to 10% on both datasets. Furthermore, these vectors consistently outperformed the `utt-mean` baseline, indicating that the speech/silence classification is important for the model. MLE-based vectors

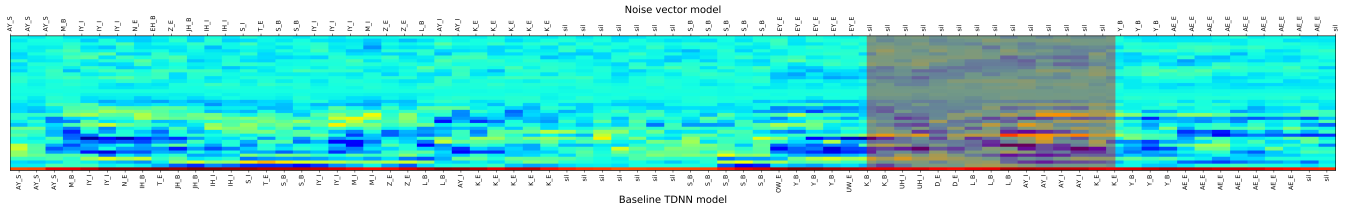

Fig. 2: Frame-level phone alignments obtained from the Baseline TDNN and noise vector models for the sentence `I MEAN IT JUST SEEMS LIKE YEAH` in the AMI Dev set. The shaded region shows the noisy silence segment where the baseline model makes insertion errors.

are also effective, although they perform slightly worse than the utterance-level vectors. We conjecture that this may be because of incorrect estimates of speech and silence in the beginning of the utterance (cf. Section 5.3).

Among the embedding-based systems from literature, i-vectors (both offline and online variants) showed substantial gains for AMI, but degraded results on Aurora-4. Similar observations were made for the NAT-vector and bottleneck NN baselines. However, e-vectors estimated using LDA were found to consistently improve WER on both datasets. Our proposed method provided the best WER results across the board, demonstrating its robustness.

5.2. Where does the improvement come from?

Table 3 shows the insertion, deletion, and substitution error rates for the proposed method. On the AMI dataset, which contains late reverberations and interfering speakers, our noise vectors reduce insertion errors substantially, suggesting that utterance-level characteristics help in speaker selection. This is further demonstrated in Fig. 2, which presents frame-level phone alignments for the baseline model and the noise-aware model for a sample utterance from the AMI dev set. We see that the baseline model inserted extraneous phonemes between `LIKE` and `YEAH`, as a result of cross-talk from the interfering speaker, but the noise-aware model correctly identified the frames as silence. Finally, the noise vectors consistently reduced deletion and insertion errors across both datasets.

5.3. Offline vs. MLE noise vectors

Fig. 3 shows 4 (out of 80) coefficients from the offline and MLE noise vectors for an utterance selected at random from the Aurora-4 `test_0166` set. We see that the online estimates for speech coefficients (15 and 35) quickly approached the utterance-level values. For the noise coefficients (55 and 75), the utterance-level averages for the online estimates were higher because of fewer silence frames in the middle of the utterance, but the overall estimate over the whole utterance was close to the offline vector. Similar observations were made across both datasets.

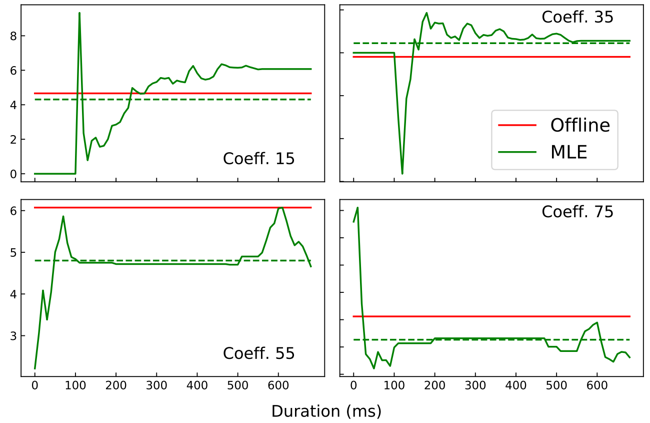

Fig. 3: Online noise estimates using MLE and MAP for a randomly selected utterance in the Aurora-4 `test_0166` set. The top and bottom row coefficients correspond to means of speech and silence frames, respectively. For each coefficient, the solid red and green lines are the "offline" and MLE estimates, respectively. The dashed green line represents the mean of the MLE noise vector over the whole utterance.

6. CONCLUSIONS

We proposed a simple method that leverages a SAD to perform noise-aware training of acoustic models in hybrid ASR systems by computing the means of speech and silence frames. Our method, which we call "noise vectors," provides significant improvements in WER in diverse conditions such as noise and far-field reverberant speech, as demonstrated through experiments conducted on Aurora-4 and AMI, respectively. On both conditions, our method outperformed strong baselines that use i-vectors, e-vectors trained with LDA, or bottleneck neural networks. Through error analysis, we hypothesized that the improvements are a results of better performance of our model on frames containing high backround noise or interference in the form of overlapping speech. Finally, we extended our method to the case of streaming ASR by showing that similar WER gains can be obtained using maximum likelihood estimates of the noise vectors. In future work, we will investigate strategies to remove the dependence on first-pass decoding by using the current decoder traceback.

7. REFERENCES

[1] K. Kinoshita, M. Delcroix, S. Gannot, E. A. Habets, R. Haeb-Umbach, W. Kellermann, V. Leutnant, R. Maas, T. Nakatani, B. Raj *et al.*, "A summary of the REVERB challenge: state-of-the-art and remaining challenges in reverberant speech processing research," *EURASIP Journal on Advances in Signal Processing*, vol. 2016, no. 1, p. 7, 2016.

[2] J. Barker, E. Vincent, N. Ma, H. Christensen, and P. Green, "The PASCAL CHiME speech separation and recognition challenge," *Computer Speech & Language*, vol. 27, no. 3, pp. 621–633, 2013.

[3] J. Barker, R. Marxer, E. Vincent, and S. Watanabe, "The third 'CHiME' speech separation and recognition challenge: Dataset, task and baselines," in *2015 IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*. IEEE, 2015, pp. 504–511.

[4] J. Li, L. Deng, Y. Gong, and R. Haeb-Umbach, "An overview of noise-robust automatic speech recognition," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 22, no. 4, pp. 745–777, 2014.

[5] S. Boll, "Suppression of acoustic noise in speech using spectral subtraction," *IEEE Transactions on acoustics, speech, and signal processing*, vol. 27, no. 2, pp. 113–120, 1979.

[6] P. C. Loizou, *Speech enhancement: theory and practice*. CRC press, 2013.

[7] C. J. Leggetter and P. C. Woodland, "Maximum likelihood linear regression for speaker adaptation of continuous density hidden markov models," *Computer speech & language*, vol. 9, no. 2, pp. 171–185, 1995.

[8] Z. Zhang, J. Geiger, J. Pohjalainen, A. E.-D. Mousa, W. Jin, and B. Schuller, "Deep learning for environmentally robust speech recognition: An overview of recent developments," *ACM Transactions on Intelligent Systems and Technology (TIST)*, vol. 9, no. 5, pp. 1–28, 2018.

[9] S. Srinivasan, N. Roman, and D. Wang, "Binary and ratio time-frequency masks for robust speech recognition," *Speech Communication*, vol. 48, no. 11, pp. 1486–1501, 2006.

[10] M. L. Seltzer, D. Yu, and Y. Wang, "An investigation of deep neural networks for noise robust speech recognition," in *2013 IEEE ICASSP*. IEEE, 2013, pp. 7398–7402.

[11] T. Higuchi, N. Ito, T. Yoshioka, and T. Nakatani, "Robust MVDR beamforming using time-frequency masks for online/offline asr in noise," in *2016 IEEE ICASSP*. IEEE, 2016, pp. 5210–5214.

[12] M. Kitza, A. Zeyer, R. Schlüter, J. Heymann, and R. Haeb-Umbach, "Robust online multi-channel speech recognition," in *Speech Communication; 12. ITG Symposium*. VDE, 2016.

[13] G. Saon, H. Soltau, D. Nahamoo, and M. Picheny, "Speaker adaptation of neural network acoustic models using i-vectors," in *2013 IEEE Workshop on Automatic Speech Recognition and Understanding*. IEEE, 2013, pp. 55–59.

[14] N. Dehak, P. Kenny, R. Dehak, P. Dumouchel, and P. Ouellet, "Front-end factor analysis for speaker verification," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 19, pp. 788–798, 2011.

[15] D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, M. Hannemann, P. Motlicek, Y. Qian, P. Schwarz *et al.*, "The Kaldi speech recognition toolkit," in *IEEE 2011 workshop on Automatic Speech Recognition and Understanding*, no. CONF. IEEE Signal Processing Society, 2011.

[16] S. Kim, B. Raj, and I. Lane, "Environmental noise embeddings for robust speech recognition," *ArXiv*, vol. abs/1601.02553, 2016.

[17] X. Feng, B. Richardson, S. Amman, and J. R. Glass, "An environmental feature representation for robust speech recognition and for environment identification," in *INTERSPEECH*, 2017.

[18] F.-H. Liu, R. M. Stern, X. Huang, and A. Acero, "Efficient cepstral normalization for robust speech recognition," in *Human Language Technology: Proceedings of a Workshop Held at Plainsboro, New Jersey, March 21-24, 1993*, 1993. [Online]. Available: https://www.aclweb.org/anthology/H93-1014

[19] K. Veselỳ, A. Ghoshal, L. Burget, and D. Povey, "Sequence-discriminative training of deep neural networks." in *INTERSPEECH*, vol. 2013, 2013, pp. 2345–2349.

[20] D. Povey, V. Peddinti, D. Galvez, P. Ghahremani, V. Manohar, X. Na, Y. Wang, and S. Khudanpur, "Purely sequence-trained neural networks for ASR based on lattice-free MMI." in *INTERSPEECH*, 2016, pp. 2751–2755.

[21] J. Rownicka, P. Bell, and S. Renals, "Embeddings for DNN speaker adaptive training," *2019 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pp. 479–486, 2019.

[22] P. Somervuo, B. Y. Chen, and Q. Zhu, "Feature transformations and combinations for improving ASR performance," in *INTERSPEECH*, 2003.

[23] J. Carletta, S. Ashby, S. Bourban, M. Flynn, M. Guillemot, T. Hain, J. Kadlec, V. Karaiskos, W. Kraaij, M. Kronenthal *et al.*, "The AMI meeting corpus: A pre-announcement," in *International workshop on machine learning for multimodal interaction*. Springer, 2005, pp. 28–39.

[24] S.-K. A. Yeung and M.-H. Siu, "Improved performance of Aurora 4 using HTK and unsupervised MLLR adaptation," in *Eighth International Conference on Spoken Language Processing*, 2004.

[25] D. Povey, G. Cheng, Y. Wang, K. Li, H. Xu, M. Yarmohammadi, and S. Khudanpur, "Semi-orthogonal low-rank matrix factorization for deep neural networks," in *INTERSPEECH*, 2018, pp. 3743–3747.

[26] M. Federico, N. Bertoldi, and M. Cettolo, "IRSTLM: an open source toolkit for handling large scale language models," in *Ninth Annual Conference of the International Speech Communication Association*, 2008.

[27] C. Cieri, D. Miller, and K. Walker, "The Fisher corpus: a resource for the next generations of speech-to-text," in *LREC*, vol. 4, 2004, pp. 69–71.

[28] A. Stolcke, "SRILM-an extensible language modeling toolkit," in *Seventh International Conference on Spoken Language Processing*, 2002.