Text-informed speech enhancement with deep neural networks

Keisuke Kinoshita[1], *Marc Delcroix*[1], *Atsunori Ogawa*[1], *Tomohiro Nakatani*[1]

[1]NTT Communication Science Labs, NTT corporation

kinoshita.k@lab.ntt.co.jp

Abstract

A speech signal captured by a distant microphone is generally contaminated by background noise, which severely degrades the audible quality and intelligibility of the observed signal. To resolve this issue, speech enhancement has been intensively studied. In this paper, we consider a text-informed speech enhancement, where the enhancement process is guided by the corresponding text information, i.e., a correct transcription of the target utterance. The proposed deep neural network (DNN)-based framework is motivated by the recent success in the text-to-speech (TTS) research in employing DNN as well as high audible-quality output signal of the corpus-based speech enhancement which borrows knowledge from the TTS research field. Taking advantage of the nature of DNN that allows us to utilize disparate features in an inference stage, the proposed method infers the clean speech features by jointly using the observed signal and widely-used TTS features derived from the corresponding text. In this paper, we first introduce the background and the details of the proposed method. Then, we show how the text information can be naturally integrated into speech enhancement by utilizing DNN and improve the enhancement performance.

Index Terms: speech enhancement, text-to-speech, deep neural network

1. Introduction

The quality and intelligibility of a speech signal captured by a distant microphone is generally degraded by acoustic interferences such as environmental noise, channel distortion and reverberation. To resolve this issue, a considerable amount of speech enhancement research has been done from various perspectives [1].

In the last decade, a great progress has been made in the field of speech enhancement research by introducing statistical speech models, such as Non-negative Matrix Factorization (NMF)-based model [2–4], Gaussian mixture model (GMM) [5] and hidden Markov model (HMM) [6]. A common objective of these methods is to accurately reproduce clean speech spectra from noisy observed spectra based on the statistical speech models. A natural and powerful extension of this type of approach has been recently introduced [7–10]. This approach, often referred to as corpus-/inventory-based approach, is greatly inspired by unit-selection-based text-to-speech synthesis (TTS) technologies [11]. The literatures [7–10] reported that very high audible-quality enhancement is possible by first looking for speech units in the training data that best matches to the underlying clean speech components in the target noisy speech, and then generating enhanced spectra by concatenating the units.

This paper investigates another way to incorporate TTS knowledge into speech enhancement. In contrast to the previous approaches [7–10], this work is motivated by model-based TTS approaches [12, 13]. More specifically, as a first step to tightly integrate enhancement and TTS technologies, this paper considers an enhancement style referred to as a *text-informed speech enhancement*, where the enhancement process is guided by the corresponding text information, i.e., a correct transcription of the target utterance. The proposed method is based on deep neural network (DNN), which is successfully but separately adopted in state-of-the-art TTS [12, 13], and speech enhancement algorithms [14–17]. In the proposed method, DNN is trained to map the input features, which comprises both noisy speech features and text features, to the output clean speech features via highly nonlinear regression. It should be noted that the style of the text-informed speech enhancement was firstly proposed in [18] based on an NMF framework. In their algorithm, first a TTS engine was used to generate a synthesized speech signal based on the input text. Then, that signal was used to guide NMF-based separation process. In general, this type of speech enhancement algorithm is categorized as informed, user-guided or user-assisted enhancement approaches as opposed to blind approaches [19]. It has potential applications such as audio postproduction, where it is quite common for audio engineers to manually guide the enhancement process by feeding certain guide signal to the system to obtain a satisfactory output.

In this paper, first we briefly review DNN-based speech enhancement as well as state-of-the-art TTS algorithms. Then, we describe the proposed method and clarify how it can be seen as a natural extension of such technologies. In an experiment, we show the effectiveness of the proposed framework in noisy environments with channel distortions.

2. Conventional methods

2.1. DNN-based speech enhancement

Approaches covered in this subsection, sometimes referred to as denoising autoencoder, perform enhancement by using a learned nonlinear mapping function between corrupted speech signals and clean speech signals as shown in the left part of Fig. 1 [14–17]. The mapping function is essentially the DNN-based regression model. Let us first denote the N-dimensional input feature vector at the l-th hidden layer as $\mathbf{x}_l$. Then, a regression function, $f_{\theta_l}(\cdot)$ ($l = 1, \ldots, L-1$), at the l-th hidden layer in DNN is given as the following affine transformation followed by a nonlinearity as:

$$f_{\theta_l}(\mathbf{x}_l) = \mathbf{x}_{l+1} = s(\mathbf{W}_l \mathbf{x}_l + \mathbf{b}_l). \quad (1)$$

$\theta_l = \{\mathbf{W}_l, \mathbf{b}_l\}$ corresponds to the parameters of the regression model. $s(\cdot)$ is called an activation function which typically takes a form of the sigmoid function, hyperbolic tangent or rectified linear unit (ReLU). As opposed to the hidden layers, a transformation from the last (i.e., L-th) hidden layer to the output nodes takes the following linear regression form:

$$g_{\theta_L}(\mathbf{x}_L) = \mathbf{y} = \mathbf{W}_L \mathbf{x}_L + \mathbf{b}_L. \quad (2)$$

As it was mentioned, the regression parameters have to be learned prior to the test stage. In the training stage, the parameters are optimized based on a collection of stereo data, consisting of pairs of corrupted and clean speech represented for example by the log-power spectra or Mel-filterbank features. Typical

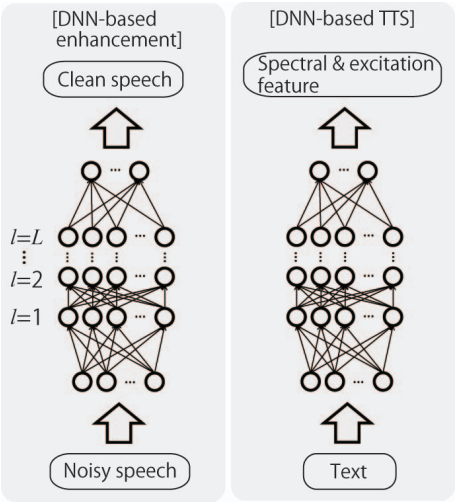

Figure 1: Diagram of the DNN-based enhancement and DNN-based TTS

optimization criterion used in DNN-based speech enhancement is minimum-mean square error (MMSE). In the test stage, the learned model is fed with the observed corrupted speech feature in order to generate the enhanced feature.

This type of method is reported to yield a very good enhanced signal, when the training and test cases are closely matched [14–17]. Note that other types of deep neural networks such as bidirectional recurrent neural networks [17] can be adopted to this framework.

2.2. DNN-based text-to-speech synthesis

The DNN framework was successfully adopted also in TTS research, and was shown to outperform the conventional decision tree-clustered context-dependent HMM-based approach [12, 13]. As shown in the right part of Fig. 1, structure of the DNN is essentially the same as the one in the previous section.

In the TTS case, the input features to the network are generated from the corresponding input text. Typical input features would include binary answers to questions about linguistic contexts (e.g., is-current-phoneme-aa?) and numeric values (e.g., the number of words in the phrase, the relative position of the current frame in the current phoneme, and durations of the current phoneme). Hereafter, these input features will be referred to as the "text features".

In the training stage, such input text features are first paired with their acoustic realizations. More specifically, the input features are paired with features such as spectral (e.g., Mel-cepstral, or line spectral pairs (LSP)) and excitation features, and their time derivatives, which will be directly used as the output features of the DNN. Then, the DNN is trained to learn a highly complex mapping function between the input and output features. In the test stage, such output features are inferred by the DNN given the input text features, and further utilized to generate speech parameters as in the conventional HMM-based synthesizer.

Generated speech signals were evaluated by means of subjective evaluation and were found to be better than the conventional methods [12, 13].

3. Proposed method: DNN-based text-informed speech enhancement

Although it has been quite difficult to utilize the text features in a speech enhancement framework previously, DNN may allow us to naturally and effectively incorporate such features. As shown in many literatures including TTS [12, 13], automatic speech recognition (ASR) [20–22] and enhancement research [23], the nature of DNN allows us to utilize disparate features jointly in an inference stage. Taking advantage of such characteristics, we here propose to incorporate the text features into the enhancement based on the DNN framework.

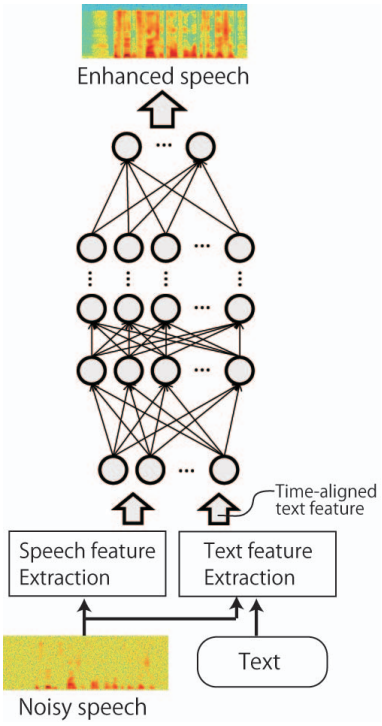

Figure 2: Text-informed speech enhancement based on a DNN

Figure 2 shows a schematic block diagram of the proposed text-informed speech enhancement method. The input features of the DNN comprise the noisy speech features (e.g., log-power-spectra or Mel-filterbank features) and the text features generated from the correct text transcription. In an actual application, this correct transcription has to be given to the system by a user. Note that, as opposed to the TTS case, now the text features has to be time-aligned to the input noisy features on a frame basis. Therefore, the text feature extraction module in Fig. 2 has to perform the following two processing: (1) extraction of the text features from the text, and (2) calculation of time alignments between the input noisy speech features and the phone sequence generated from the text by using a forced alignment function [24] within an ASR system [1]. As the text features, we consider the following 3 types of features used in the conventional TTS [12, 13].

(TF1) Binary answers to the set of phonetic questions [25] (e.g., is-current-phoneme-vowel?) about the current quinphone identity (i.e., identities about the current phoneme, the previous phoneme, the next phoneme, the phoneme before the previous phoneme, and the phoneme after the next phoneme) [2].

(TF2) Duration of the current phoneme and relative position of the current frame in the current phoneme.

[1] Such forced alignment is carried out quite regularly for example in the ASR acoustic model training stage.

[2] Regarding the phoneme before the previous phoneme and the phoneme after the next phoneme, only a broad phonetic category was described in the feature vector.

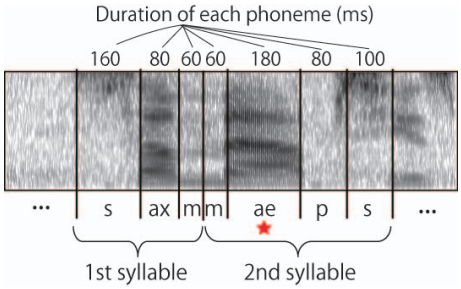

Figure 3: Forced alignment result of the input text "some maps..." and the input noisy signal saying "some maps...".

(TF3) The number of phonemes in the current syllable and the position of the current phoneme in the syllable.

Figure 3 shows an example of the result of the forced alignment of the input text "some maps..." and the input noisy signal saying "some maps...". With the forced alignment, we can obtain exact position and duration of the phonemes contained in the utterance "some maps...". Now, if we would like to extract the text features of the frame marked with ★, we immediately know that the current quinphone is m-m-ae-p-s, and thus (TF1) should indicate this identity in a binary-value vector form. Likewise, (TF2) should contain the information about the duration of the phoneme "ae", i.e., 180 ms, and current frame position within this phoneme, i.e., 90 ms. (TF3) should essentially contain 4 and 2, because the current phoneme "ae" is in the syllable "m-ae-p-s" consisting of 4 phonemes, and is the second one in the syllable.

The output features of the DNN of the proposed method consists of clean speech feature such as clean log-power-spectra, clean Mel-filterbank features, or ones typically used in a DNN-based TTS system. To synthesize the output waveform, we can combine the enhanced feature with the phase information taken from the observed noisy speech. In case that the output feature is Mel-filterband features, we can first convert them back to the linear frequency scale, form a Wiener filter and perform noise reduction. In case that the output feature is log-power spectra, we can either directly convert it back to the waveform with the noisy phase, or form a Wiener filter and perform noise reduction.

4. Experiment

To show the effectiveness of the proposed method, we evaluate its performance in comparison with the conventional DNN-based speech enhancement (i.e., denoising autoencoder) introduced in Section 2.1.

4.1. Experimental conditions

We used Aurora-4 database to carry out a large scale evaluation in noisy conditions with channel distortions. Aurora-4 is based on the Wall Street Journal 5k task, comprising about 14 hours of speech for the training set including 83 speakers. To generate the training data, 6 different types of noise (street traffic, train stations, cars, babble, restaurants, airports) were artificially added to clean speech or speech containing channel distortion, at randomly selected SNRs between 10 and 20 dB. To train the DNN of the conventional and the proposed method, noisy training data and the corresponding clean speech data were used. Note that for monitoring the convergence of the DNN learning process, we randomly extracted 5% of the training data for a so-called validation set, and used the remaining 95% of the data for the actual training. Since the training data contains various speakers, the trained DNN would become speaker independent.

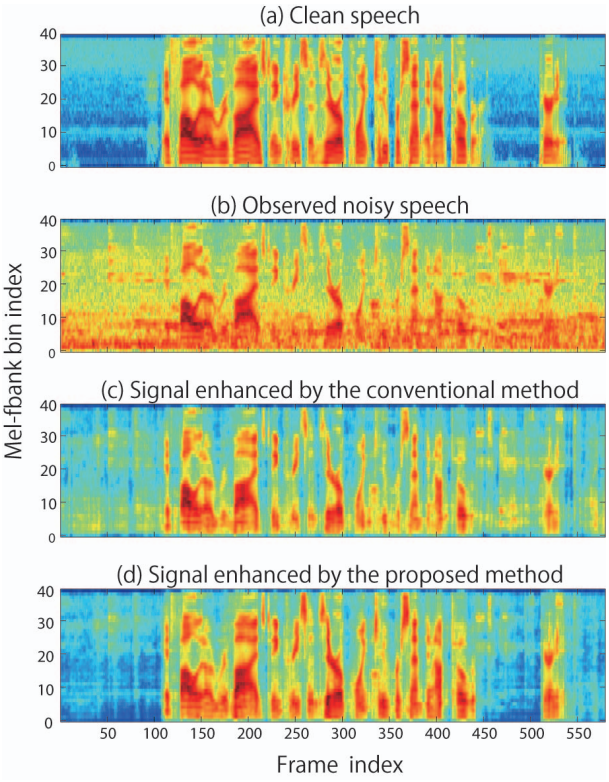

Figure 4: Log Mel-filterbank feature of (a) clean speech, (b) noisy observed speech, (c) signal estimated by the conventional method, (d) signal estimated by the proposed method (prop-2), in case of "babble" noise

For testing the algorithms, we used the development set of Aurora-4 which contains the same types of noise but with SNRs of between 5 and 15 dB. The sampling frequency of the data was 16kHz.

4.2. Details regarding the features and parameters

For both the conventional and the proposed methods, we considered the standard feed-forward DNN structure. The network was first descriminatively pretrained [26], where we train the whole network every time we add one more hidden layer. And then, it was fine-tuned with back-propagation with the MMSE criterion. To examine the effect of the depth of the network, the number of hidden layers was changed from 1 to 5. The number of nodes in hidden layers was set to 2048 and that of the output node was set to 40, which corresponds to the vector dimension of the static log Mel-filterbank feature of the current frame.

For the input feature of the conventional method, 40-order Log Mel-filterbank feature and its 1st and 2nd derivatives were used [3]. Following the common practice in speech processing, the feature of the current frame was spliced with features within 5 left and 5 right context frames to form an input feature vector consisting of 11 frames. As the speech feature of the proposed method, we used the same speech feature as the conventional method. As the text feature, we used (TF1), (TF2) and (TF3) features summarized in Section 3. The feature (TF1), (TF2), and (TF3) comprises 329-dimentional binary-value vector, 2-dimentional numeric-value vector, and 5-dimentional numeric-

[3] We also tried log-power-spectral features, and obtained similar results

(a) Conditions w/o channel distortion

	clean	car	babble	restaurant	street	airport	train
No proc	**0.00**	4.26	4.00	4.08	4.40	3.89	4.41
Conventional	0.90	1.66	2.13	2.30	2.14	2.04	2.14
Prop-1	0.86	1.58	1.93	2.12	1.98	1.84	1.99
Prop-2	0.83	**1.57**	**1.91**	**2.11**	**1.97**	**1.82**	**1.98**
Prop-3	0.90	1.62	1.96	2.17	2.02	1.88	2.04

(b) Conditions w/ channel distortion

	clean	car	babble	restaurant	street	airport	train
No proc	3.65	4.92	4.59	4.58	4.82	4.58	4.86
Conventional	2.07	2.27	2.72	2.85	2.65	2.60	2.63
Prop-1	1.99	2.16	2.45	2.59	2.45	2.35	2.44
Prop-2	**1.98**	**2.14**	**2.44**	**2.58**	**2.42**	**2.33**	**2.43**
Prop-3	2.02	2.18	2.48	2.63	2.47	2.38	2.48

Figure 6: Cepstral distance (dB) of the unprocessed signal, the conventional method (5-layer case), and each of the proposed methods (5-layer case) under each noise condition without channel distortion (upper panel) and with channel distortion (lower panel). The best score at each condition is marked with a bold font.

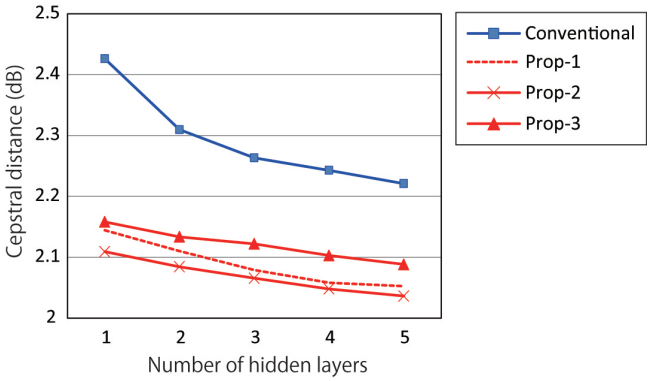

Figure 5: Cepstral distance of the conventional method, and proposed methods as a function of the number of hidden layers. Cepstral distance between clean speech and unprocessed observed speech was 4.07 dB.

value vector, respectively. Numeric features in the text features as well as the speech features were normalized to have zero mean and unit variance before feeding them to the network. To examine the effectiveness of the text features in more details, we prepared 3 variations of the proposed method, where "Prop-1", "Prop-2" and "Prop-3" correspond to the ones taking the input feature of the speech feature + (TF1) + (TF2) + (TF3), the speech feature + (TF1) + (TF2), the speech feature + (TF1), respectively. The forced alignment in the text feature extraction module was carried out using a GMM-HMM ASR recognizer trained on the multi-condition data.

4.3. Results

Figure 4 shows the log-Mel filterbank feature of (a) clean speech, (b) corresponding noisy speech observed in babble noise, (c) signal estimated by the conventional method, (d) signal estimated by the proposed method (prop-2 with 5 hidden layers). First of all, it is clearly shown that an appropriate inference of clean speech was achieved by the proposed method. Although the conventional method also works quite reasonably even in severe noisy conditions, estimation errors are clearly visible for example as remaining energy in silence regions, and some overly suppressed speech components. On the other hand, the proposed method could manage to suppress the remaining noise very effectively, as well as somewhat recover the missing speech features.

Figure 5 shows the cepstral distance between the clean log Mel-filterbank features and the enhanced signals as a function of the number of hidden layers. Note that the cepstral distance between the clean log Mel-filterbank features and the unprocessed observed signal was 4.07 dB. We can first confirm that all the variations of the proposed methods outperform significantly the conventional method, in each condition. From the figure, it can be also seen that the deeper network structures are more beneficial for the enhancement.

Figure. 6 shows the details of the cepstral distance at each noisy and channel distortion condition. It can be seen that, although the test signal contains 7 different types of noisy conditions including clean condition and 2 types of channel distortions, in all cases the proposed methods outperformed the conventional method.

Audible quality of the enhanced signal is also improved, but the resultant signal still contains some remaining noise and artifacts. Such noise and artifacts can be reduced or removed for example by changing the network topology, estimating other features than log Mel-filterbank features, or reconstruct the enhanced signal in a different manner. Such investigations are within the scope of our future work. It should be also noted that, although the learned DNN is speaker independent, the signal directly reconstructed from the enhanced feature clearly maintained the corresponding speaker identity.

5. Conclusions

A speech signal captured by a distant microphone is generally contaminated by background noise, which severely degrades the audible quality and intelligibility of the observed signal. Among numerous studies on speech enhancement, recently it was found that the knowledge in TTS research field could potentially benefit speech enhancement technology greatly especially when high audible-quality is desirable. Toward a rigorous integration of speech enhancement and TTS technologies, this paper investigated a way to incorporate text information into the enhancement processing, by utilizing the DNN framework. It exploits auxiliary text features that are provided by a user to guide the enhancement process. In the proposed method, speech features and text features are jointly used as an input feature of DNN, and are mapped to the output clean speech features via highly nonlinear DNN regression. It was shown that the estimated log Mel-filterbank features are quite close to that of the reference clean speech, and improved the quality of speech enhancement.

6. References

[1] X. Huang, A. Acero, and H.-W. Hon, *Spoken language processing*. Upper Saddle River, NJ: Prentice Hall, 2001.

[2] K. Wilson, B. Raj, P. Smaragdis, and A. Divakaran, "Speech de-noising using nonnegative matrix factorization with priors," in *ICASSP*, 2008, pp. 4029 – 4032.

[3] P. Smaragdis, "Convolutive speech bases and their application to supervised speech separation," *IEEE Trans. Audio Speech Lang. Process.*, vol. 15(1), pp. 1 – 12, 2007.

[4] N. Mohammadiha, P. Smaragdis, and A. Leijon, "Supervised and unsupervised speech enhancement using nonnegative matrix factorization," *IEEE Trans. Audio Speech Lang. Process.*, vol. 21(10), pp. 2140–2151, 2013.

[5] P. Moreno, "Speech recognition in noisy environments," *PhD thesis*, 1996.

[6] T. Kristjansson, J. Hershey, P. Olsen, and R. Gopinath, "Super-human multi-talker speech recognition: The IBM 2006 speech separation challenge system," in *Proc. Int'l Conf. Speech and Language Process. (ICSLP)*, 2006, pp. 97–100.

[7] X. Xiao, P. Lee, and R. M. Nickel, "Inventory based speech enhancement for speaker dedicated speech communication systems," in *ICASSP*, 2009, pp. 3877–3880.

[8] J. Ming, R. Srinivasan, and D. Crookes, "A corpus-based approach to speech enhancement from nonstationary noise," *IEEE TASLP*, vol. 19, no. 4, pp. 822–836, 2011.

[9] A. Ogawa, K. Kinoshita, T. Hori, T. Nakatani, and A. Nakamura, "Fast segment search for corpus-based speech enhancement based on speech recognition technology," in *ICASSP*, 2014, pp. 1557–1561.

[10] M. I. Mandel, Y.-S. Cho, and Y. Wang, "Learning a concatenative resynthesis system for noise suppression," in *IEEE GlobalSIP conference*, 2014, pp. 582–586.

[11] A. J. Hunt and A. W. Black, "Unit selection in a concatenative speech synthesis system using a large speech database," in *ICASSP*, vol. 1, 1996, pp. 373–376.

[12] H. Zen, A. Senior, and M. Schuster, "Statistical parametric speech synthesis using deep neural networks," in *ICASSP*, 2013, pp. 7962 – 7966.

[13] Y. Fan, Y. Qian, F. Xie, and F. K. Soong, "TTS synthesis with bidirectional LSTM based recurrent neural networks," in *ICASSP*, 2014, pp. 1964–1968.

[14] A. Maas, Q. Le, T. O ' Neil, O. Vinyals, P. Nguyen, and A. Ng, "Recurrent neural networks for noise reduction in robust ASR," in *Interspech*, 2012.

[15] Y. Xu, J. Du, L.-R. Dai, and C.-H. Lee, "An experimental study on speech enhancement based on deep neural networks," *IEEE Signal Process. Letters*, vol. 21(1), pp. 65 – 68, 2014.

[16] K. Han, Y. Wang, and D. L. Wang, "Learning spectral mapping for speech dereverberation," in *ICASSP*, 2014, pp. 4661–4665.

[17] F. Weninger, S. Watanabe, Y. Tachioka, and B. Schuller, "Deep recurrent de-noising auto-encoder and blind dereverberation for reverberated speech recognition," in *ICASSP*, 2014, pp. 4656–4659.

[18] L. L. Magoarou, A. Ozerov, and N. Q. K. Duong, "Text-informed audio source separation using nonnegative matrix partial co-factorization," in *IEEE International Workshop on Machine Learning for Signal Processing*, 2013.

[19] A. Liutkus, J.-L. Durrieu, and L. D. an G. Richard, "An overview of informed audio source separation," in *2013 14th International Workshop on Image Analysis for Multimedia Interactive Services (WIAMIS)*, 2013, pp. 1–4.

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

[21] G. Saon, H. Soltau, D. Nahamoo, and M. Picheny, "Speaker adaptation of neural network acoustic models using i-vectors," in *ASRU*, 2013, pp. 55 – 59.

[22] A. Narayanan and D. L. Wang, "Joint noise adaptive training for robust automatic speech recognition," in *ICASSP*, 2014, pp. 2523–2527.

[23] S. Araki, T. Hayashi, M. Delcroix, M. Fujimoto, K. Takeda, and T. Nakatani, "Exploring multi-channel features for denoising-autoencoder-based speech enhancement," in *ICASSP*, 2015, p. to appear.

[24] P. Moreno, C. Joerg, J.-M. V. Thong, and O. Glickman, "A recursive algorithm for the forced alignment of very long audio segments," in *ICSLP*, 1998.

[25] X. Huang, A. Acero, and H.-W. Hon, *Spoken language processing*. Upper Saddle River, NJ: Prentice Hall, pp.435.

[26] F. Seide, G. Li, X. Chen, and D. Yu, "Feature engineering in context-dependent deep neural networks for conversational speech transcription," in *ASRU*, 2011, pp. 24–29.