Robust Automatic Speech Recognition for Call Center Applications

Luis Felipe Parra-Gallego[1,2(✉)], Tomás Arias-Vergara[1,3], and Juan Rafael Orozco Arroyave[1,3]

[1] GITA Lab. Faculty of Engineering, University of Antioquia UdeA, Medellín, Colombia
{lfelipe.parra,tomas.arias,rafael.orozco}@udea.edu.co
[2] Konecta Group S.A.S., Medellín, Colombia
[3] Pattern Recognition Lab, Friedrich-Alexander-Universität Erlangen-Nürnberg, Erlangen, Germany

Abstract. This paper is focused on developing an Automatic Speech Recognition (ASR) system robust against different noisy scenarios. ASR systems are widely used in call centers to convert telephone recordings into text transcriptions which are further used as input to automatically evaluate the Quality of the Service (QoS). Since the evaluation of the QoS and the customer satisfaction is performed by analyzing the text resulting from the ASR system, this process highly depends on the accuracy of the transcription. Given that the calls are usually recorded in non-controlled acoustic conditions, the accuracy of the ASR is typically decreased. To address this problem, we first evaluated four different hybrid architectures: (1) Gaussian Mixture Models (GMM) (baseline), (2) Time Delay Neural Network (TDNN), (3) Long Short-Term Memory (LSTM), and (4) Gated Recurrent Unit (GRU). The evaluation is performed considering a total of 478,6 h of recordings collected in a real call-center. Each recording has its respective transcription and three perceptual labels about the level of noise present during the phone-call: Low level of noise (LN), Medium Level of noise (ML), and High Level of noise (HN). The LSTM-based model achieved the best performance in the MN and HN scenarios with $22,55\%$ and $27,99\%$ of word error rate (WER), respectively. Additionally, we implemented a denoiser based on GRUs to enhance the speech signals and the results improved in 1,16% in the HN scenario.

Keywords: ASR · Noise reduction · Speech enhancement · Speech-to-text

1 Introduction

Millions of calls answered in call centers are recorded and stored every day. The recordings are used for different purposes including to improve the Quality of

Supported by University of Antioquia.

J. C. Figueroa-García et al. (Eds.): WEA 2021, CCIS 1431, pp. 72–83, 2021.
https://doi.org/10.1007/978-3-030-86702-7_7

the Service (QoS). Typically the QoS evaluation process consists in listening to the conversations between call-center agents and customers to label whether the service requested by the customer was successfully provided or not. This procedure is usually done by humans who evaluate the service by randomly taking samples from the total set of calls. During the evaluation process, it is analyzed the reason for the call, the emotional state of both the customer and advisor, the effectiveness and promptness of the service provided by the agent, and others [18]. Although the aforementioned is the standard procedure, it has two main disadvantages: (1) it is very expensive and time consuming, and (2) only a few samples over the total calls are evaluated, so it is not possible to know about all critical calls that could negatively impact the service [18].

With the aim to make the above mentioned process more efficient, automatic systems are designed to rate the calls based on the text transcription of the spoken conversation. This is performed by using Automatic Speech Recognition (ASR) systems. Once the conversations are transcribed, several information are extracted from the resulting texts including keywords, key sentences, number and types of hesitations, specific expressions, and others. The main advantage of ASR-based systems is that they enable the analysis of all answered calls automatically. Although ASR systems are the natural way to go in order to improve QoS in call centers, their accuracy directly affects the performance of the system that rates the calls (the one that is based on text analysis). This means that transcriptions with errors could produce wrong interpretations for the QoS evaluation system. Typically, an ASR works with high performance under ideal acoustic conditions; however, many different factors reduce the ASR performance, such as the speaker's health condition and emotional state, the communication channel including the microphone and the sound card, environmental noise, and others.

This work aims of developing ASR system robust against different noisy scenarios. We first evaluate four hybrid architectures in three levels of noise: low, medium, and high. We then propose to implement a Deep Learning based-denoiser in order to clean the speech signal and thus improve the recognition performance. The denoiser is based on Complex Linear Coding (CLC), a similar approach presented in [13]. To assess the denoising technique, we re-evaluated the ASR systems' performance when passing the noisy recordings through the filter.

The rest of this paper is organized as follows: Sect. 2 presents an overview of related works; Sect. 3 describes the database; Sect. 4 introduces the methodology followed in this work; Sect. 5 shows the results obtained in this study; and Sect. 6 includes the conclusions and future work.

2 Related Works

Several techniques have been proposed in the literature to model acoustic features in ASR systems. The most typical approaches used nowadays are those based on Hidden Markov Models - Gaussian Mixture Models (HMM-GMM),

and Hidden Markov Models - Deep Neural Network (HMM-DNN) [16]. HMM-GMM has played an important role in designing conventional recognizers because they are easy to train and have low computational cost [16]. Thanks to the advances in computational power and machine learning algorithms, DNN has shown excellent results in different applications including ASR. In [5] and [4], results using different acoustic models in ASR systems with different acoustic conditions are reported. Both works show that DNN-based models outperform classical methods based on GMMs. Different topologies of networks have been proposed to improve the ASR performance. In [14] three topologies are compared: (1) Recurrent Neural Network, (2) Long Short-Term Memory (LSTM), and (3) Gated Recurrent Unit (GRU). The authors used a total of 378 audio recordings from the TED talks in English. The dataset contains files for training, validation, and test. Spectrograms were used to train the acoustic model and the best WER was achieved using LSTM (65.04%). The authors reported that GRU showed similar results (67,42% WER) in a shorter period training time; GRU only ran for 5 days and 5 h while LSTM required slightly more than seven days.

More complex architectures based on end-to-end systems have been recently proposed. In [17], the authors compared different "very deep models". Convolutional LSTM with a residual connection (reConvLSTM) was also introduced in the same work. Convolutional LSTM layers basically replace multiplication operations among parameters and inputs by convolutions. Their architecture consists of 2 convolutional layers, followed by 4 ResConvLSTM and finally an LSTM Network in Network block. A total of 80 filter-banks with their deltas were used as the feature set. The Wall Street Journal (WSJ) English corpus [8] was used to train and test the network. This database contains 73 h for training and 8 h for testing. The model proposed by Zhang et al. in [17] showed a WER of 10,53%, while previous studies were around 18% in the same corpus. In the same line, the authors in [1] proposed an end-to-end system where its input is the raw speech signal. To do that, they used a convolutional filter learning based on rectangular band-pass filters. This technique is called SincNet. The authors proposed to connect SincNet to an end-to-end recurrent encoder-decoder structure using joint CTC-attention procedure. It was used WSJ corpus [8], and TIMIT corpus [3] for training and testing the model. The authors compared their system with traditional end-to-end models operating on Mel-filter-banks. For the TIMIT database, their technique did not show improvements in comparison to conventional hybrid DNN-HMM perhaps due to the small amount of available training data (less than 5 h). On the other hand, when using WSJ, their technique obtained a top-of-line WER of 4.5%, outperforming all baselines. The previous best score was 5.9% WER, which means an absolute improvement of 1.2%.

Other kinds of techniques as speech enhancement, domain adaptation, and data augmentation have also been studied with DNNs. In [7], it was proposed the problem-agnostic speech encoder (PASE), a novel architecture that combines a convolutional encoder followed by multiple neural networks, called workers, tasked to solve self-supervised problems. The aim of each worker is to generate

features extracted from the original speech signal as MFCCs, log power spectrum, gammatone features, waveform speech signal, and others. The needed consensus across different tasks naturally imposed meaningful constraints to the encoder, contributing to discover general representations and hence minimizing the risk of learning superficial features. The authors performed self-supervised training with a portion of 50 h of the LibriSpeech dataset [5]; TIMIT [3], DIRHA [11] and CHiME-5 [2]. In order to validate the technique, the authors trained a hybrid DNN-HMM speech recognizer using different acoustic features such as MFCC, filter bank, gammatone, and MFCC + gammatone + filter bank. The features extracted from the PASE architecture significantly outperformed the other feature sets, with a relative improvement of 9.5% in clean speech and of 17.7% in noisy conditions using TIMIT.

3 Databases

We evaluate the proposed ASR systems using a call-center database called KONECTADB. The recordings were captured in non-controlled acoustic conditions, so we implemented a denoising technique to enhance the speech signals. We split the dataset into two parts, train and test, with the aim to optimize and evaluate the ASR system and the denoiser. The augmented version of KONECTADB was created by adding noise of the Demand Noise Dataset (DND).

3.1 KONECTADB

This corpus contains recordings of conversations between customers and agents of a contact-center of the Konecta Group company in Medellín, Colombia. The customers were informed that their speech was going to be recorded. Due to the nature of the service, it is assumed that the speakers in these recordings are all of legal age. The database consists of 478, 6 h of audio with a sampling frequency of 8 kHz and a 16 bits resolution. Experts in QoS annotated the recordings in the contact center. Each audio has its transcription text, the customer's gender, and its level of noise. Since the recordings were captured in non-controlled acoustic environments, this database is useful to evaluate the robustness of ASR systems against noisy conditions. Table 1 shows the demographic information of this database.

3.2 Demand Noise Dataset (DND)

The DND corpus [15] contains a variety of noise signals taken in real-world acoustic environments. The database considers two scenarios, namely "inside" environments and "open-air" environments. The inside recordings are divided into Domestic, Office, Public, and Transportation; while the open-air recordings are classified as Street and Nature. All recordings are captured with a 16-channel array of microphones sampling at 48 kHz. Thus, each environment noise recording is actually a set of 16 mono sound files.

Table 1. Demographic description of Konecta database. **LN:** Low level of noise. **MN:** Medium level of noise. **HN:** High level of noise. **Male:** Number of male recordings. **Female:** Number of female recordings

Label of noise	# of speakers	Gender distribution		Hours	
		Male	Female	Training	Test
LN	18938	19459	27313	321,0	30,3
MN	6615	7180	8191	101,4	15,9
HN	633	636	666	8,7	1,2

3.3 Data Augmentation

Clean recordings of KONECTADB are augmented by adding noise signals of the DND corpus. The noisy samples are created by randomly taking two different noises from the DND corpus associated with different Signal-to-Noise Ratio (SNR) levels: $-5, 0, 5, 10, 20,$ and 40 dB. To achieve the selected SNR level, the noise is scaled by a factor α, which is expressed as:

$$\alpha = \sqrt{\frac{P_{s(t)}}{SNR \cdot P_{n(t)}}} \tag{1}$$

where $P_{s(t)}$, $P_{n(t)}$, and SNR are speech signal power, noise signal power, and SNR computed in linear scale, respectively.

Training and test sets were augmented separately and used to train and evaluate the denoiser described in Sect. 4.1. The data augmentation algorithm is depicted in Fig. 1.

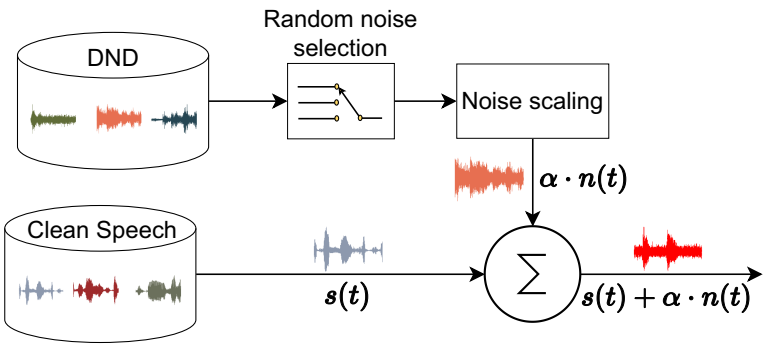

Fig. 1. Data augmentation process.

4 Methodology

Figure 2 illustrates the overall process to train and test an ASR system. At the top, it is described the training stage, and at the bottom the test one.

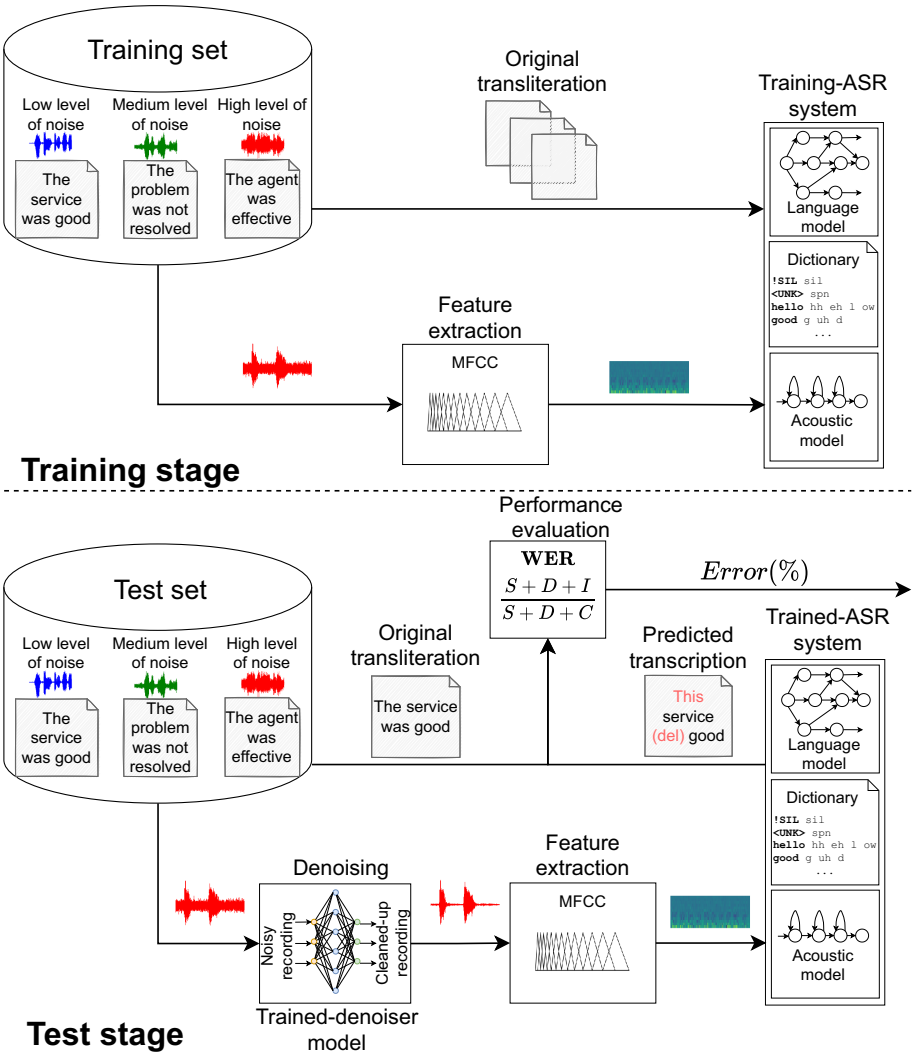

Fig. 2. General methodology followed in this study.

4.1 Training Stage

This stage encompasses feature extraction, Language Model (LM), Acoustic Model (AM), and the Dictionary.

4.1.1 Feature Extraction

This study considered a total of 40-MFCCs extracted from 40 triangular Mel-frequency bins with a window size of 25 ms and a step size of 10 ms. The spectrogram is unit-normalized.

4.1.2 Language Model

The transliteration of the training set was used to train a 3-gram language model. The probabilities of a language model can be computed by counting relative frequencies of the 3−tuples of words that belong to the training set. To estimate the probabilities of the 3−gram model, the following equation is used:

$$P(w_n|w_{n-1}, w_{n-2}) = \frac{C(w_{n-2}, w_{n-1}, w_n)}{C(w_{n-2}, w_{n-1})} \tag{2}$$

where w_n represents a word located in position n, and C represents a function that counts the number of occurrences of the word sequence defined in its argument.

4.1.3 Acoustic Model

This study considers a 3-state HMM for modeling temporal dependencies. We trained and evaluated four different models to represent the acoustic distribution of each acoustic unit (HMM state).

- **GMM:** This acoustic model is based on GMM models. A total of 100 thousand of Gaussian components and a decision tree of 4016 leaves were considered in this work. The GMMs were trained using a Maximum Likelihood estimation. This model was also used to force-align the training data and is regarded as the baseline in this study.
- **TDNN:** This architecture consists of six TDNN layers with 1536 units and a bottleneck dimension of 256. Each layer contains a frame context of three and a skip connection coming from the previous layer's input. The last TDNN layer's output is fed into a fully connected layer with a softmax activation function. Details of this method can be found in [9].
- **LSTM:** This architecture consists of four bidirectional LSTM layers with a *tanh* activation function. Each layer contains 550 units and a dropout regularization of 0.2. The last LSTM layer's output is fed into a fully connected layer with a $softmax$ activation function.
- **GRU:** This architecture consists of five bidirectional GRU layers with a *relu* activation function. Each layer contains 550 units and a dropout regularization of 0.2. The last GRU layer's output is fed into a fully connected layer with a $softmax$ activation function.

The forced-aligned data generated by the GMMs were used to train the DL-based models. On the one hand, the Kaldi toolkit [10] was used to train the TDNN model using Stochastic Gradient Decent (SGD) with an initial learning rate of 0.00015 and batch size of 64. On the other hand, ADAM optimizer with an initial learning rate of 0.0002 and batch size of 64 was used to train the LSTM- and GRU- based architectures using Pytorch-Kaldi framework [12]. We only considered five epochs due to computational constraints.

4.1.4 Dictionary

The dictionary contains the phone pronunciation of each word to be recognized in our model. The phone composition is performed using pronunciation rules of the Spanish language from Colombia. To build the dictionary, the most frequent words seen in the training set were selected. This study considered 20 thousand different words.

4.2 Test Stage

This involves the same processes as the training stage and also includes denoising and performance evaluation. To avoid any possible bias and to guarantee the generalization capability of the model, this process only considers recordings of the test set.

4.2.1 Denoising Model

The denoiser is thought to enhance the speech signals. We used a similar approach as the one presented in [13]. A Short-Time Fourier transform (STFT) was computed using a 25 ms Hanning window with a step size of 10 ms. The model architecture consists of a fully connected layer followed by two GRU layers and finally, two fully connected layers. The input to the model is the unit-normalized complex spectrogram. The last layer predicts the masking coefficient to denoise the complex spectrogram. The mask aims to reduce noise effects by multiplying weights closer to zero with those frequency bands that contain noise energy. The masked complex spectrogram is then transformed into the time-domain using the inverse STFT function. The complete filter process is illustrated in Fig. 3.

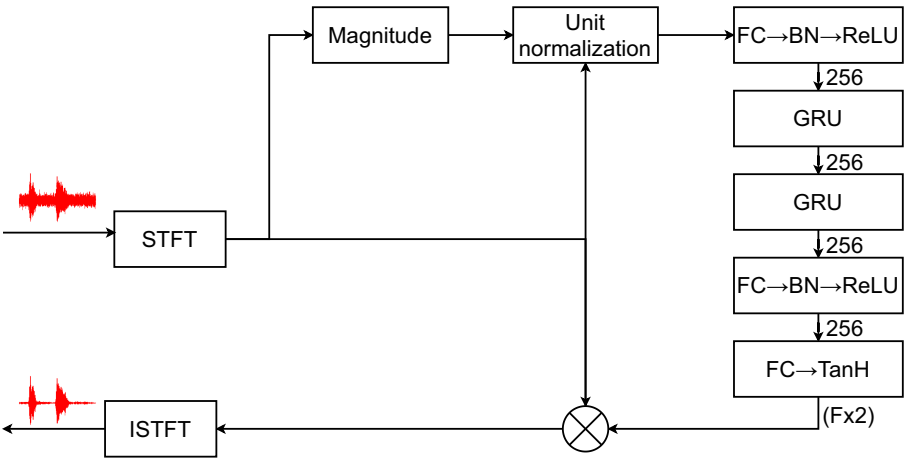

Fig. 3. Denoising process [13]. F is the number of frequency bins and $\otimes$ is the Hadamard product.

The augmented training dataset of KONECTADB described in Sect. 3 is used to train the system. The original signals are used as the ground truth during the training process of the GRU. The GRU-based denoiser is trained with Pytorch using the Adam optimization strategy with an initial learning rate of 0.0001 and a batch size of 10. Only five epochs are considered due to computational constraints.

4.2.2 Performance Evaluation

Once the ASR system is trained, this is used to convert the recordings into text transcriptions of the test set. Word Error Rate (WER) was computed to evaluate the model. This is the a well known performance measure typically used to evaluate ASR systems [6]. It is defined as follows:

$$WER = \frac{S + D + I}{S + D + C} \tag{3}$$

where,

- S: # of substitutions.
- D: # of deletions.
- I: # of insertions.
- C: # of correctly recognized words.

WER compares two text chains. This metric counts the number of operations needed to convert one text into another one. WER is computed upon the original transcription and the predicted transcription in the case of an ASR system.

5 Results

With the aim to develop a robust ASR system, we trained and evaluated four different acoustic model architectures in non-controlled acoustic scenarios. The following are the models: (1) GMM-based model, (2) TDNN-based model, (3) LSTM-based model, and (4) GRU-based model. Finally, a DL-based denoiser is implemented to improve the recognition performance.

5.1 Results of Acoustic Model

The call center database described in Sect. 3 is used to train each ASR system. The LN, MN, and HM training sets were mixed during the training. The models are evaluated in each real acoustic scenario. Table 2 shows the performance of the different ASR systems for each scenario. Note that all DL-based models outperformed the baseline (based on GMMs). The LSTM model yields the best performance in non-controlled acoustic conditions with WER values of $22,55\%$ and $27,99\%$ for MN and AN scenarios, respectively. Note that all models except the GMM-based one, obtained similar WER values in the LN condition, that is: $21,73\%$ for TDNN, $21,31\%$ for LSTM, and $21,30$ for GRU.

Table 2. Performance of the ASR systems in terms of the WER in each real acoustic conditions. **LN:** Low level of noise. **MN:** Medium level of noise. **HN:** High level of noise.

Architecture	Acoustic scenario		
	LN	MN	HN
GMM	32,10	35,54	52,47
TDNN	21,73	23,48	30,94
LSTM	**21,31**	**22,55**	**27,99**
GRU	**21,30**	22,67	28,77

5.2 Results of Denoising Process

The denoiser described in Sect. 4.1 was trained to enhance noisy speech signals. The augmented training dataset of KONECTADB was used to train the model. Two test sets are considered to evaluate the capability of the filter to suppress/remove the noise: (1) The artificially created noisy recordings (The scenario described in Sect. 3), and (2) The noisy recordings of the HN test set (Real scenario). WER values of the ASR systems are computed for the noisy and enhanced speech signals for comparison purposes. Table 3 shows the performance obtained for the DL-based ASR systems in the simulated and real scenarios. The TDNN model shows improvements in both scenarios when the denoiser is applied. In the simulated conditions, the WER goes down from 40,41% to 35,70%, and in the real noisy conditions it changes from 30,94% to 26,83% which is actually the best performance obtained for noisy conditions. For the case of the LSTM-based model in the simulated scenario, without denoising it yields the worst WER for noisy conditions (44,39%), but it improves to 38,88% after applying the denoiser. Although the improvement is relatively high (5,51 absolute percentage points), the result is still the worst among the rest obtained in that scenario. Regarding its results in the real conditions, without any denoising procedure, the LSTM yields the best WER (27,99%), however, when the denoiser is applied the WER value increases to 29,63%. A similar behavior can be observed for the GRU model, where the WER value obtained in the simulated conditions prior to the denoiser is 40,41% and it gets better to 37,27% when the denoiser is applied; however, in the real noisy conditions, its WER value gets worst in 1 absolute percentage when the denoiser is applied (from 28,77% to 29,77%).

Table 3. Performance of the ASR systems in terms of the WER before and after applying the denoiser. **Simulated:** The augmented test set. **Real:** The HN test set of KONECTADB. Values in %.

Model	Simulated conditions		Real conditions	
	Noisy	Enhanced	Noisy	Enhanced
TDNN	**40,41**	**35,70**	30,94	**26,83**
LSTM	44,39	38,88	**27,99**	29,63
GRU	40,41	37,27	28,77	29,77

6 Conclusions and Future Work

This work presented a methodology to improve the recognition performance of ASR systems. We trained and evaluated four different acoustic models in non-controlled acoustic conditions: (1) GMM-based model (Baseline), (2) TDNN-based model, (3) LSTM-based model, and (4) GRU-based model. The models were trained with recordings of a call center database, called KONECTADB. This database contains customer service telephone calls. Each recording was captured in real acoustic conditions and it was labeled in terms of its level of noise: low, medium and high. These acoustic conditions allowed us to evaluate the models in real noisy acoustic conditions. The LSTM-model achieved the best performance for medium and high levels of noise, which likely indicates that this model is the most robust in non-controlled conditions.

With the aim to improve the recognition performance, a DL-based filter was developed to clean the speech signals. The portion of KONECTADB with low level of noise was artificially contaminated with noise signals taken from Demand Noise Dataset. We trained the denoiser using the noisy recordings. Once the denoiser was trained, the ASR models were again evaluated in two scenarios: (1) Simulated (The artificially contaminated test set), and (2) the real test set with the recordings originally labeled as high level of noise. The WER was computed before and after passing the recordings through the denoiser. On the one hand, the performance of the GRU- and LSTM- based models decreased after the denoising process. But, on the other hand, the TDNN model achieved the best results when the denoiser was applied in both simulated and real acoustic scenarios. The observed improvement was 1,16% of WER with respect to the result obtained by the LSTM without any denoising strategy. For future work we consider to explore more complex architectures in the denoising process to see whether the performance can be further improved.

Acknowledgements. This work was funded by Konecta Group in association with CODI at the University of Antioquia, grant # PI2019-24110. Tomás Arias-Vergara is under grants of Convocatoria Doctorado Nacional-785 financed by MINCIENCIAS.

References

1. Agrawal, P., Ganapathy, S.: Interpretable filter learning using soft self-attention for raw waveform speech recognition. arXiv preprint arXiv:2001.07067 (2020)
2. Barker, J., Watanabe, S., Vincent, E., Trmal, J.: The fifth 'CHiME' speech separation and recognition challenge: dataset, task and baselines. arXiv preprint arXiv:1803.10609 (2018)
3. Garofolo, J.S., Lamel, L.F., Fisher, W.M., Fiscus, J.G., Pallett, D.S., Dahlgren, N.L.: Timit acoustic-phonetic continuous speech corpus LDC93S1. Web Download. Linguistic Data Consortium, Philadelphia (1993)
4. Kinoshita, K., et al.: The REVERB challenge: a benchmark task for reverberation-robust ASR techniques. In: Watanabe, S., Delcroix, M., Metze, F., Hershey, J.R. (eds.) New Era for Robust Speech Recognition, pp. 345–354. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-64680-0_15

5. Panayotov, V., Chen, G., Povey, D., Khudanpur, S.: Librispeech: an ASR corpus based on public domain audio books. In: 2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 5206–5210. IEEE (2015)
6. Park, Y., Patwardhan, S., Visweswariah, K., Gates, S.C.: An empirical analysis of word error rate and keyword error rate. In: Ninth Annual Conference of the International Speech Communication Association (2008)
7. Pascual, S., Ravanelli, M., Serrà, J., Bonafonte, A., Bengio, Y.: Learning problem-agnostic speech representations from multiple self-supervised tasks. arXiv preprint arXiv:1904.03416 (2019)
8. Paul, D.B., Baker, J.M.: The design for the wall street journal-based CSR corpus. In: Proceedings of the Workshop on Speech and Natural Language, pp. 357–362. Association for Computational Linguistics (1992)
9. Povey, D., Cheng, G., Wang, Y., Li, K., Xu, H., Yarmohammadi, M., Khudanpur, S.: Semi-orthogonal low-rank matrix factorization for deep neural networks. In: Interspeech, pp. 3743–3747 (2018)
10. Povey, D., et al.: The kaldi speech recognition toolkit. Technical report, IEEE Signal Processing Society (2011)
11. Ravanelli, M., Cristoforetti, L., Gretter, R., Pellin, M., Sosi, A., Omologo, M.: The Dirha-English corpus and related tasks for distant-speech recognition in domestic environments. In: 2015 IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU), pp. 275–282. IEEE (2015)
12. Ravanelli, M., Parcollet, T., Bengio, Y.: The pytorch-kaldi speech recognition toolkit. In: ICASSP 2019–2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 6465–6469. IEEE (2019)
13. Schröter, H., Rosenkranz, T., Maier, A., et al.: CLC: complex linear coding for the DNS 2020 challenge. arXiv preprint arXiv:2006.13077 (2020)
14. Shewalkar, A., Nyavanandi, D., Ludwig, S.A.: Performance evaluation of deep neural networks applied to speech recognition: RNN, LSTM and GRU. J. Artif. Intell. Soft Comput. Res. **9**(4), 235–245 (2019)
15. Thiemann, J., Ito, N., Vincent, E.: Demand: a collection of multi-channel recordings of acoustic noise in diverse environments. In: Proceedings of Meetings Acoust (2013)
16. Yu, D., Deng, L.: Automatic Speech Recognition: A Deep Learning Approach. Springer, London (2015). https://doi.org/10.1007/978-1-4471-5779-3
17. Zhang, Y., Chan, W., Jaitly, N.: Very deep convolutional networks for end-to-end speech recognition. In: 2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 4845–4849. IEEE (2017)
18. Zweig, G., et al.: Automated quality monitoring in the call center with ASR and maximum entropy. In: 2006 IEEE International Conference on Acoustics Speech and Signal Processing Proceedings, vol. 1, p. I. IEEE (2006)