Minimum Word Error Training of RNN-based Voice Activity Detection

Gregory Gelly[*†], *Jean-Luc Gauvain*[*]

[*]LIMSI - CNRS, Spoken Language Processing Group, Orsay, France
[†]Paris-Sud University, Orsay, France
gelly@limsi.fr, gauvain@limsi.fr

10.21437/Interspeech.2015-565

Abstract

Voice Activity Detection (VAD) is critical in speech recognition systems as it can dramatically impact the recognition accuracy especially on noisy data. This paper presents a novel method which applies Minimum Word Error (MWE) training to a Long Short-Term Memory RNN to optimize Voice Activity Detection for speech recognition. Experiments compare speech recognition WERs using RNN VAD with other commonly used VAD methods for two corpora: the conversational Vietnamese corpus used in the NIST OpenKWS13 evaluation and a corpus of French telephone conversations. The proposed VAD method combining MWE training with RNN yields the best ASR results. This MWE training scheme appears to be particularly useful for low resource ASR tasks, as exemplified by the IARPA BABEL data.

Index Terms: speech recognition, minimum word error, voice activity detection, recurrent neural networks, long short-term memory, particle swarm optimization

1. Introduction

Voice Activity Detection (VAD) is a crucial task in any speech processing system. Concerning ASR systems, it directly impacts the accuracy as too much speech undetected speech will result in undesired deletions, and too much non-speech labelled as speech can increase the number of insertions and will also slow down the decoding with unnecessary processing.

A variety of methods and models have been proposed exploiting the spectro-temporal properties of speech and noise to effectively separate speech from non-speech. Some of these methods are energy-based [1, 2], others use auto-correlation coefficients [3, 4] or features that describe the degree of non-stationarity of the signal over long window frames (200-300ms) to successfully discriminate noise from noisy speech signal [5].

Neural Network (NN) based methods have also been proposed both to provide VAD features [6, 7, 8] and to offer a higher-level decision making mechanism [9, 10].

Two innovations to improve VAD performance for ASR are introduced. We proposed a new optimization framework based on Minimum Word Error to optimize VAD for speech recognition. This is applied to a new recurrent neural networks specifically designed for the VAD task.

To validate our Minimum Word Error training, a comparitive study with 5 different VAD methods is performed:

- *CrossCorr*: a feature based on the maximum peak of the normalized autocorrelation function of the signal [4] ;
- *LTSV*: the Long-Term Signal Variability proposed by Ghosh et al. [11] ;
- and 3 different NN-based methods described in the next section.

2. NN-based VAD

In this paper, we focus on three different types of neural networks that were used for VAD where the single output of the NNs is used as a confidence score that the current frame is speech. For the 3 proposed models (MLP, BLSTM, BLSTM+) we used MFCCs (including delta features) as input. In order to have a fair comparison between the 3 types of neural networks, all were configured to have the same number of weights (6000). It was found that increasing the number of weights did not improve the performance.

2.1. Multi-Layer Perceptron (*MLP*)

To each input vector $\boldsymbol{p}$ (e.g. cepstral coefficients) the classic *fully connected feed-forward network with one hidden layer* associates an output vector $\boldsymbol{z}$ computed as follows:

$$\boldsymbol{z} = \sigma_z \left(\boldsymbol{W}_z \cdot \sigma_h \left(\boldsymbol{W}_h \cdot \boldsymbol{p} + \boldsymbol{b}_h \right) + \boldsymbol{b}_z \right) \quad (1)$$

where $\boldsymbol{W}_h$ and $\boldsymbol{W}_z$ are the *interconnection matrices* (or *weight matrices*) of the network, $\boldsymbol{b}_h$ and $\boldsymbol{b}_z$ are the *bias vectors* of the network, σ_h and σ_z are the *transfer functions* of the network. The latter are typically chosen among bounded non-linear functions such as the hyperbolic tangent applied element-wise.

2.2. Bidirectional Long Short-Term Memory (*BLSTM*)

To make the most of the context around each audio frame, we used recurrent neural networks (RNN) based on Long Short-Term Memory cells as shown in Figure 1. LSTM cells were introduced to overcome some of the shortcomings of classic RNNs [12] and were popularized by A. Graves for their good performance on optical character recognition and speech sequence labelling tasks [13, 14].

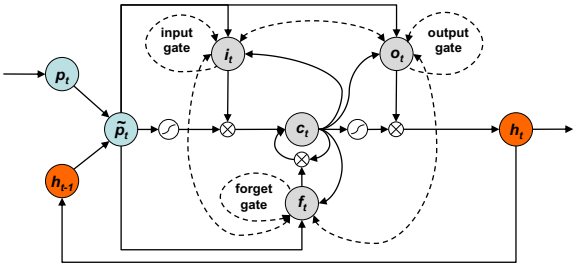

Figure 1: *LSTM cell. The dashed lines correspond to the added links between the gates for the augmented LSTM cell.*

Given an input sequence $\boldsymbol{p} = (\boldsymbol{p}^1, ..., \boldsymbol{p}^T)$, a standard RNN computes the output vector sequence $\boldsymbol{z} = (\boldsymbol{z}^1, ..., \boldsymbol{z}^T)$ by iter-

ating the following equations from $t = 1 \rightarrow T$:

$$\boldsymbol{h}^t = \sigma_1\left(\boldsymbol{W}_1 \cdot \tilde{\boldsymbol{p}}^t + \boldsymbol{b}_1\right) \quad with \quad \tilde{\boldsymbol{p}}^t = \begin{bmatrix} \boldsymbol{p}^t \\ \boldsymbol{h}^{t-1} \end{bmatrix} \tag{2}$$

$$\boldsymbol{z}^t = \sigma_z\left(\boldsymbol{W}_z \cdot \boldsymbol{h}^t + \boldsymbol{b}_z\right) \tag{3}$$

The use of LSTM cells instead of the classic summation units modifies the computation of $\boldsymbol{h}^t$ as follows:

$$\boldsymbol{i}^t = \sigma_i\left(\boldsymbol{W}_i \cdot \tilde{\boldsymbol{p}}^t + \boldsymbol{W}_i^c \cdot \boldsymbol{c}^{t-1} + \boldsymbol{b}_i\right) \tag{4}$$

$$\boldsymbol{f}^t = \sigma_f\left(\boldsymbol{W}_f \cdot \tilde{\boldsymbol{p}}^t + \boldsymbol{W}_f^c \cdot \boldsymbol{c}^{t-1} + \boldsymbol{b}_f\right) \tag{5}$$

$$\boldsymbol{c}^t = diag\left(\boldsymbol{f}^t\right) \cdot \boldsymbol{c}^{t-1} + diag\left(\boldsymbol{i}^t\right) \cdot \sigma_c\left(\boldsymbol{W}_c \cdot \tilde{\boldsymbol{p}}^t + \boldsymbol{b}_c\right) \tag{6}$$

$$\boldsymbol{o}^t = \sigma_o\left(\boldsymbol{W}_o \cdot \tilde{\boldsymbol{p}}^t + \boldsymbol{W}_o^c \cdot \boldsymbol{c}^t + \boldsymbol{b}_o\right) \tag{7}$$

$$\boldsymbol{h}^t = diag\left(\boldsymbol{o}^t\right) \cdot \sigma_h\left(\boldsymbol{c}^t\right) \tag{8}$$

where $\boldsymbol{i}^t$, $\boldsymbol{f}^t$, $\boldsymbol{c}^t$ and $\boldsymbol{o}^t$ are respectively the *input gate*, the *forget gate*, the *cell* and the *output gate* activation vectors. They are all the same size as the hidden vector $\boldsymbol{h}^t$. $\boldsymbol{W}_i^c$, $\boldsymbol{W}_f^c$, $\boldsymbol{W}_o^c$ are diagonal matrices so that each heart of a cell is only visible to the gates of the same cell.

One shortcoming of conventional RNNs is that they are only able to make use of past context. For VAD purposes there is no reason not to exploit future context as well. Bidirectional LSTM neural networks (*BLSTM*) were developed to do just that: 2 distinct LSTM networks process the sequence both forward and backward, and then the output of both networks are combined and fed into the output layer. This way, we can fully exploit the long range capabilities of LSTM cells. In the literature (e.g. [13, 14]) BLSTM networks always perform better than unidirectional ones, therefore we explored only BLSTM networks in this study.

2.3. Augmented BLSTM

We propose a modified version of the BLSTM neural network in which direct links are added between the three gates of a LSTM cell as shown by the dashed lines in Figure 1. This modfication aims to prevent that some of the LSTM cells get stuck in a saturated state when trained on long sequences. The equations (4), (5) and (7) are thus modified into (10), (12) and (14):

$$\tilde{\boldsymbol{i}}^t = \boldsymbol{W}_i^i \cdot \boldsymbol{i}^{t-1} + \boldsymbol{W}_i^f \cdot \boldsymbol{f}^{t-1} + \boldsymbol{W}_i^o \cdot \boldsymbol{o}^{t-1} \tag{9}$$

$$\boldsymbol{i}^t = \sigma_i\left(\boldsymbol{W}_i \cdot \tilde{\boldsymbol{p}}^t + \boldsymbol{W}_i^c \cdot \boldsymbol{c}^{t-1} + \tilde{\boldsymbol{i}}^t + \boldsymbol{b}_i\right) \tag{10}$$

$$\tilde{\boldsymbol{f}}^t = \boldsymbol{W}_f^i \cdot \boldsymbol{i}^{t-1} + \boldsymbol{W}_f^f \cdot \boldsymbol{f}^{t-1} + \boldsymbol{W}_f^o \cdot \boldsymbol{o}^{t-1} \tag{11}$$

$$\boldsymbol{f}^t = \sigma_f\left(\boldsymbol{W}_f \cdot \tilde{\boldsymbol{p}}^t + \boldsymbol{W}_f^c \cdot \boldsymbol{c}^{t-1} + \tilde{\boldsymbol{f}}^t + \boldsymbol{b}_f\right) \tag{12}$$

$$\tilde{\boldsymbol{o}}^t = \boldsymbol{W}_o^i \cdot \boldsymbol{i}^t + \boldsymbol{W}_o^f \cdot \boldsymbol{f}^t + \boldsymbol{W}_o^o \cdot \boldsymbol{o}^{t-1} \tag{13}$$

$$\boldsymbol{o}^t = \sigma_o\left(\boldsymbol{W}_o \cdot \tilde{\boldsymbol{p}}^t + \boldsymbol{W}_o^c \cdot \boldsymbol{c}^t + \tilde{\boldsymbol{o}}^t + \boldsymbol{b}_o\right) \tag{14}$$

where the nine matrices $\boldsymbol{W}_{\{i,f,o\}}^{\{i,f,o\}}$ are diagonal so that a gate can only have access to the gates of the same cell.

With these new links the three gates of a cell can interact more efficiently and improve the behavior of the cell. We name this new network *BLSTM+*.

3. VAD Smoothing

In this work, the output of all VAD methods are post-processed with the same smoothing technique, the parameters of which are optimized during the VAD training for each method. The smoothing parameters are:

- an onset and offset thresholds for the detection of the beginning and end of a speech segment;
- padding durations before and after each speech segment;
- a threshold for short speech segment deletion;
- and a threshold for small silence deletion.

4. Minimum Word Error Training

Since our goal is to develop a better VAD method for speech recognition, we designed the following Minimum Word Error (MWE) training that would allow us to optimize the behavior of the VAD to minimize the WER. This method was applied using the NIST OpenKWS13 conditions, that we should only make use of the language-specific provided data and no previously existing models.

4.1. Loss functions

Optimizing a VAD algorithm to minimize the WER of an ASR system would be best achieved by computing the actual WER for all the VAD settings tested by the optimization algorithm. Unfortunately, the computational load is too high for it to be a realistic solution. Instead three loss functions related to the WER are use to optimize the VAD model parameters with the aim to minimize the WER. The only assumption here is that we have available a complete training data set, i.e. the audio files and their orthographic transcriptions.

4.1.1. Frame error rate wrt the human reference (L_1)

The first loss function is the frame error rate (FER) with respect to manual annotations and is defined as follows:

$$L_1 = \alpha \sum_{s \in S} \delta_s(z) + (1-\alpha) \sum_{n \in N} \delta_n(z) \tag{15}$$

where S is the set of speech frames, N is the set of non-speech frames, z is the binary output of the VAD, $\delta_s(z)$ equals 0 except if z equals 0, $\delta_n(z)$ equals 0 except if z equals 1 and α sets the relative importance between errors on the speech frames and errors on the noise frames. To minimize the FER, one has to set α to 0.5. But to minimize the WER it is more important to minimize the number of missed speech frames than the FER per say. Therefore different values for α were tried and the best value was found to be 0.6 on our data.

An advantage of this loss function is that it does not require a prior ASR system. This is particularly useful when starting from scratch as required by the NIST OpenKWS rules.

4.1.2. Frame error rate wrt the ASR output (L_2)

When the ASR system is available, its output (words, time-codes, confidence scores) can be used as a reference for VAD training. More precisely, the correct words and the substitutions are considered as speech, whereas the silences, the deletions and the insertions are considered as non-speech. It was decided to treat the deletions as non-speech because we had no way of accurately knowing where the boundaries of the deleted words were and we wanted a precise speech tagging. In this framework, the form of the loss function stays the same, only

the set of speech frames S and the set of non-speech frames N are modified. This loss functions is called L_2.

Using this metric the best value for α was found to be 0.85 on our data. The difference with L_1 resides in the confidence that we have in the tagging of speech frames. Human annotators often tag small in-between silences as speech whereas the frames corresponding to correct words and substitutions in the ASR output can reliably be considered as speech. It is then no surprise that a higher α is best suited in this case. Note that we still need to allow for some false alarms to take into account the deletions of the ASR system.

4.1.3. WER-like metric (L_3)

The L_2 loss function makes use of the ASR output but does not take into account the main specificity of the WER metric, i.e. every word long or short, substitution, insertion or deletion weighs the same in the final performance. We thus designed a loss function L_3 that reflects this.

Let W_C, W_S, W_D and W_I respectively be the sets of correct words, substitutions, deletions and insertions in the ASR output when compared to the human reference. For each word $w \in W_C \cup W_S \cup W_I$, we denote F_w the set of audio frames corresponding to w in the ASR output. Then we define pW_S, pW_D and pW_I as follows:

- a word $w \in pW_S$ if and only if $w \in W_S$ and all frames in F_w are classified as speech by the VAD.
- a word $w \in pW_D$ if and only if either $w \in W_D$ or $w \in W_C \cup W_S$ and at least one frame in F_w is classified as non-speech by the VAD.
- a word $w \in pW_I$ if and only if $w \in W_I$ and at least one frame in F_w is classified as speech by the VAD.

The following two ratios are also introduced:

- τ_d^w is the ratio between the duration of the word w not detected as speech and the whole duration of the word.
- τ_i^w is the ratio between the duration of the word w detected as speech and the whole duration of the word.

Finally the loss function is defined as

$$L_3 = \frac{pS + pD + pI + \tau_i + \tau_d}{N_{words}} \tag{16}$$

$$with \quad \tau_i = \sum_{w \in pW_I} \tau_i^w \quad and \quad \tau_d = \sum_{w \in W_C \cup W_S} \tau_d^w \tag{17}$$

where pS, pD and pI are respectively the numbers of words in pW_S, pW_D and pW_I. The two terms τ_i and τ_d were introduced to smooth the discontinuities of the WER metric. As a direct result, the optimization algorithm is less prone to being trapped on a plateau of the loss function.

4.2. Minimization algorithms

The interest of optimization techniques such as Genetic Algorithms for minimizing complex loss functions with an important number of parameters (> 10) was shown in [15]. Since then, similar but more efficient methods such as Quantum-behaved Particle Swarm Optimization (QPSO) were developed ([16], [17] and [18]). QPSO is a variant of the PSO algorithm which was motivated by the social behavior of bird flocks and was first introduced by Kennedy and Eberhart as a population-based optimization technique ([19] and its well known variant from Clerc [20]).

Although the PSO algorithm is comparable in performance with the Genetic Algorithms approach, QPSO proved to be a more powerful tool than both of them when performing difficult optimization tasks. All the results presented here were obtained using QPSO for training the different VAD algorithms. For the NN-based VADs, QPSO is used to simultaneously optimize the MFCC parameters, the neural network weights and the final smoothing parameters.

In addition, we found that using a second optimization technique specific to neural networks (the RPROP backpropagation algorithm [21]) is beneficial when used to locally improve the best solution found by QPSO (see section 5.1). For the BLSTM neural networks, back-propagation through time was used as described in [22] and its LSTM version [13].

Since the back-propagation algorithms need differentiable loss functions, we use a weighted version of the maximum likelihood loss function of a binary classifier when using L_1 or L_2 for QPSO. When using L_3, we designed a differentiable equivalent to eq. 16:

$$\begin{aligned} L_{3b} = &- \sum_{w \in W_C \cup W_S} \left(\frac{1}{\delta_w} \sum_{f \in w} \ln(z_f) \right) \\ &- \sum_{w \in W_I} \left(\frac{1}{\delta_w} \sum_{f \in w} \ln(1 - z_f) \right) \end{aligned} \tag{18}$$

where $z_f \in [0; 1]$ is the output of the neural network for the frame f and δ_w is the number of frames in the word w.

5. Experimental Results

Experiments were carried out using two conversational telephone speech corpora: the Vietnamese corpus used in the NIST OpenKWS13 evaluation and a large corpus in French. These corpora were chosen since they provide varied conditions in size and recognition performance.

For both languages state-of-the-art ASR systems were used. Similar to the systems described in [23], these two systems used MLP acoustic models and 4-gram language models with vocabularies of 7K words for Vietamese and 72K words for French.

A multilingual VAD developed at the end of the 90's [24] based on MFCC features and Gaussian Mixture Models (GMMs) was used as a baseline.

5.1. Vietnamese results

The IARPA Babel Program Vietnamese language collection release babel107b-v0.7 is approximately gender-balanced and contains a diversity of styles, speakers and environments. The training set includes 160 hours of audio for 87 hours of speech. The development set consists of 20 hours of audio for 11 hours of speech, and the evaluation set 30 hours of audio for 17 hours of speech.

The 5 VAD models were optimized on the training set using the the loss functions described in section 4.1. The ASR performance was measured on the development and evaluation sets. The results with the optimized VADs are given in Table 1.

While the "classic" VADs (CrossCorr and LTSV) give good performance and are easy to optimize, they were outperformed by the NN-based VADs. Among these both BLSTM VADs perform better the MLP VAD with BLSTM+ giving the best results.

With the BLSTM+ VAD, the WER of the ASR system is reduced by 2.5 points (4.4% relative) compared to the multilingual baseline. As shown in Table 2, this gain comes mainly from reducing the number of insertions (6.4% down to 3.6%). Indeed, the BLSTM+ uses its intrinsic flexibility to efficiently learn to discard signal (even speech) that lead the ASR to insert unnecessary words.

	Development set			Evaluation set		
	L_1	L_2	L_3	L_1	L_2	L_3
Baseline	53.4			57.2		
CrossCorr	53.0	52.8	52.8	56.9	56.4	56.1
LTSV	53.0	52.7	52.1	56.2	56.0	55.6
MLP	52.7	52.5	52.1	56.2	55.7	55.4
BLSTM	52.5	52.1	51.7	55.8	55.4	54.8
BLSTM+	52.4	52.0	**51.4**	55.7	55.3	**54.6**

Table 1: *WER on the development and evaluation sets of the IARPA Babel Vietnamese corpus.*

Note that the behavior of the different VADs is the same on both the development and the evaluation sets, showing the robustness of the proposed approach and of the VADs themselves. Moreover, of the loss functions, L_3 proves especially relevant since it yields the best results for all VAD algorithms.

	WER	Corr	Subs	Del	Ins
Baseline	57.2	49.2	34.7	16.1	6.4
CrossCorr	56.1	48.8	34.4	16.8	5.0
LTSV	55.6	48.9	33.5	17.6	4.5
MLP	55.4	49.0	34.2	16.8	4.4
BLSTM	54.8	49.1	34.0	16.9	3.9
BLSTM+	**54.6**	**49.0**	**34.1**	**16.9**	**3.6**

Table 2: *Detailed results of the best settings for each algorithm on the Vietnamese evaluation set.*

To evaluate the relative importance of the QPSO and the back-propagation algorithms different combinations of these two minimization algorithms have been tested for the BLSTM+ model using L_3. Table 3 reports the resulting WER on the evaluation set. These results show that both algorithms are needed to achieve the best performance when used in alternating fashion.

Optimization process using L_3	WER
Baseline	57.2
BackProp	58.0
BackProp+QPSO(smoothing)	57.5
QPSO	56.9
Hybrid QPSO-BackProp	55.5
QPSO+BackProp	55.0
QPSO+BackProp+QPSO(smoothing)	**54.6**

Table 3: *Impact of the optimization process on the performance (IARPA Babel Vietnamese evaluation data set).*

5.2. French results

The French corpus is made of telephone conversations from native and non native speakers. The VAD training set includes 290 hours of audio for 154 hours of speech, and the VAD evaluation set includes 8 hours of audio for 4 hours of speech. The CrossCorr, LTSV and the BLSTM+ VADs were optimized on the training set using the L_3 loss function. The performance of the ASR system (trained on about 1000h of data) for the evaluation data with the optimized VADs is given in Table 4.

At the time, the multilingual baseline VAD system was partly trained on this French training data set and the ASR was trained using its output. As a result, the multilingual VAD performs better than CrossCor and LTSV on the French evaluation data set. But as for the Vietnamese, the BLSTM+ model gives the best results with a slightly better WER.

	WER	Corr	Subs	Del	Ins
Baseline	23.7	78.6	13.6	7.8	2.3
CrossCorr	24.1	78.4	13.8	7.7	2.5
LTSV	23.8	78.7	13.8	7.6	2.4
BLSTM+	**23.6**	**78.9**	**13.7**	**7.5**	**2.4**

Table 4: *Detailed performance on the French evaluation set for the various VADs when optimized with the L_3 loss function.*

6. Conclusion

We have proposed a new method to optimize voice activity detection systems with the goal of minimizing the WER of a downstream ASR system. This method, which relies on the QPSO algorithm to minimize a WER-like loss function (L_3), was shown to be very effective to estimate the parameters of various VAD models, including models based on auto-correlation coefficients, a long-term signal variability measure, as well as MLP and RNN neural networks. Experimental results have been reported using two conversational speech corpora, i.e. the Vietnamese corpus of the NIST OpenKWS13 evaluation and a French corpus. The best results, measured in term of WER, are obtained using a modified bidirectional long short-term memory RNN (BLSTM+). This VAD model outperforms the other models trained on the same language specific data, as well as a multilingual GMM VAD system which served as a baseline. The BLSTM network also comes with a high decoding speed (0.001xRT) measured on a standard desktop CPU.

In the future, it would be interesting to see if further improvements could be obtained by retraining the ASR system using the BLSTM+ VAD as input.

7. Acknowledgments

This work was partially supported by the French National Agency for Research as part of the SALSA (Speech And Language technologies for Security Applications) project under grant ANR-14-CE28-0021 and by the Intelligence Advanced Research Projects Activity (IARPA) via Department of Defense US Army Research Laboratory contract number W911NF-12-C-0013. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright annotation thereon. Disclaimer: The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of IARPA, DoD/ARL, or the U.S. Government.

8. References

[1] L. Lamel, L. Rabiner, A. E. Rosenberg, and J. G. Wilpon, "An improved endpoint detector for isolated word recognition," *Acoustics, Speech and Signal Processing, IEEE Transactions on*, vol. 29, no. 4, pp. 777–785, 1981.

[2] G. Evangelopoulos and P. Maragos, "Speech event detection using multiband modulation energy." in *INTERSPEECH*, 2005, pp. 685–688.

[3] T. Kristjansson, S. Deligne, and P. Olsen, "Voicing features for robust speech detection," *Entropy*, vol. 2, no. 2.5, p. 3, 2005.

[4] H. Ghaemmaghami, B. J. Baker, R. J. Vogt, and S. Sridharan, "Noise robust voice activity detection using features extracted from the time-domain autocorrelation function," *Proceedings of Interspeech 2010*, 2010.

[5] J. Ramırez, J. C. Segura, C. Benıtez, A. De La Torre, and A. Rubio, "Efficient voice activity detection algorithms using long-term speech information," *Speech communication*, vol. 42, no. 3, pp. 271–287, 2004.

[6] T. Ng, B. Zhang, L. Nguyen, S. Matsoukas, X. Zhou, N. Mesgarani, K. Veselỳ, and P. Matejka, "Developing a Speech Activity Detection System for the DARPA RATS Program." in *INTERSPEECH*, 2012.

[7] I. Shafran and R. Rose, "Robust speech detection and segmentation for real-time ASR applications," in *Acoustics, Speech, and Signal Processing, 2003. Proceedings.(ICASSP'03). 2003 IEEE International Conference on*, vol. 1. IEEE, 2003, pp. I–432.

[8] G. Saon, S. Thomas, H. Soltau, S. Ganapathy, and B. Kingsbury, "The IBM speech activity detection system for the DARPA RATS program." in *INTERSPEECH*, 2013, pp. 3497–3501.

[9] T. Hughes and K. Mierle, "Recurrent neural networks for voice activity detection," in *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*. IEEE, 2013, pp. 7378–7382.

[10] Y. K. Thomas Drugman, Yannis Stylianou and M. Akamine, "Voice Activity Detection: Merging Source and Filter-based Information," *IEEE Signal Processing Letters*.

[11] P. K. Ghosh, A. Tsiartas, and S. Narayanan, "Robust voice activity detection using long-term signal variability," *Audio, Speech, and Language Processing, IEEE Transactions on*, vol. 19, no. 3, pp. 600–613, 2011.

[12] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[13] A. Graves, *Supervised sequence labelling with recurrent neural networks*. Springer, 2012, vol. 385.

[14] A. Graves, A.-R. Mohamed, and G. Hinton, "Speech recognition with deep recurrent neural networks," in *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*. IEEE, 2013, pp. 6645–6649.

[15] G. Gelly and P. Vernis, "Neural Networks as a Guidance Solution for Soft-Landing and Aerocapture," in *AIAA Guidance, Navigation, and Control Conference Chicago, Illinois*, 2009.

[16] J. Sun, B. Feng, and W. Xu, "Particle swarm optimization with particles having quantum behavior," in *Congress on Evolutionary Computation*, 2004.

[17] J. Sun, X. Wu, V. Palade, W. Fang, C.-H. Lai, and W. Xu, "Convergence analysis and improvements of quantum-behaved particle swarm optimization," *Information Sciences*, vol. 193, pp. 81–103, 2012.

[18] X. Fu, W. Liu, B. Zhang, and H. Deng, "Quantum behaved particle swarm optimization with neighborhood search for numerical optimization," *Mathematical Problems in Engineering*, vol. 2013, 2013.

[19] R. C. Eberhart and J. Kennedy, "A new optimizer using particle swarm theory," in *Proceedings of the sixth international symposium on micro machine and human science*, vol. 1. New York, NY, 1995, pp. 39–43.

[20] M. Clerc and J. Kennedy, "The particle swarm-explosion, stability, and convergence in a multidimensional complex space," *Evolutionary Computation, IEEE Transactions on*, vol. 6, no. 1, pp. 58–73, 2002.

[21] M. Riedmiller and H. Braun, "A direct adaptive method for faster backpropagation learning: The RPROP algorithm," in *Neural Networks, 1993., IEEE International Conference on*. IEEE, 1993, pp. 586–591.

[22] P. J. Werbos, "Backpropagation through time: what it does and how to do it," *Proceedings of the IEEE*, vol. 78, no. 10, pp. 1550–1560, 1990.

[23] V.-B. Le, L. Lamel, A. Messaoudi, W. Hartmann, J.-L. Gauvain, C. Woehrling, J. Despres, and A. Roy, "Developing STT and KWS systems using limited language resources," 2014.

[24] J. Gauvain, L. Lamel, and G. Adda, "Partitioning and transcription of broadcast news data," in *ICSLP98 5th International Conference on Spoken Language Processing, 30th November - 4th December, Sydney, Australia, Proceedings*, 1998, pp. 1335–1338.