NASAL SPEECH SOUNDS DETECTION USING CONNECTIONIST TEMPORAL CLASSIFICATION

Milos Cernak and Sibo Tong

Idiap Research Institute, Martigny, Switzerland

Milos.Cernak@ieee.org, Sibo.Tong@idiap.ch

ABSTRACT

Phone attributes, known also as distinctive or phonological features, belong to important classification of the speech sounds used in automatic speech processing. Training of conventional phone attribute detectors (classifiers), either based on acoustic measurements or deep learning approaches, requires decent phone boundary segmentation.

This paper proposes a solution to train a phone attribute detector without phone alignment using an end-to-end phone attribute modeling based on the connectionist temporal classification. Experiments, performed for the nasal phone attribute on the LibriSpeech database, confirm that the proposed system outperforms conventional deep neural network detector, trained even on the same training data. Further improvements are observed with more training data. Conventional complex system that consists of feature extraction, phone force-alignment and deep neural network training is replaced by a more simpler Python package based on PyTorch, released as open-source.

Index Terms— Phone attributes, nasal sounds, connectionist temporal classification

1. INTRODUCTION

Speech communication is known to be an asynchronous process due to asynchronous evolution of various articulatory feature streams [1]. For example, syllable stress impacts nasality of neighbouring vowels in /ini/ articulation [2]. Velopharyngeal closure during the vowel /i/ is more enhanced during stressed syllables. The unstressed co-articulated /i/ thus might convey more nasality than unstressed one, and the nasal phone attribute is asynchronous to the phone [n].

The best current phone attribute detectors are based on Deep Neural Networks (DNNs) [3, 4] that require phone alignment to get target training labels. Firstly, this does not solve asynchronous relation of the phone and phone attributes, and secondly, it makes the training process more complex as decent acoustic models have to be trained before the DNN training.

This paper proposes a solution to train a phone attribute detector without phone alignment. The solution differs from all previous approaches that required phone alignment either for bootstrapping of the DNN detector, or delineate nasal regions for acoustic measurements [5, 6, 7]. The proposed solution is based on an end-to-end phone attribute modeling aiming to simplify the system blocks into a single network architecture and matching the asynchronous relation of the phone and phone attributes. There are two major types of end-to-end architectures, the connectionist temporal classification (CTC) approach [8] and the attention-based method [9]. The latter exploits an attention mechanism to perform alignment between acoustic features and recognized symbols. However, the basic temporal attention mechanism is too flexible in the sense that it allows extremely non-sequential alignments. This is rational for applications such as machine translation where input and output word order are different [10]. However in phone attribute detection, the acoustic features and the corresponding outputs proceed in a monotonic way. Since CTC permits an efficient computation of a strictly monotonic alignment using dynamic programming, we propose to train a CTC-based phone attribute detector.

Experiments are performed on the nasals (the speech sounds with the nasal phone attribute). Nasality is phenomenon in speech processing, and it has been studied since 1956 [11]. Nasals are the only class of sounds with dominant speech output from nasal cavity instead of the oral cavity [12]. This gives them some very unique properties and makes nasals one of the hardest sounds to study. There are many complex characteristics of nasals, especially in the time evolution of the spectral domain. In addition, linear predictive coding models the vowels as an all-pole linear system, except for nasalized vowels, which have the main nasal resonance paired with an antiresonance (pole-zero pairs).

The main addressed question of this work was if it is possible to train a CTC detector without phone alignment that performs as good as a DNN detector trained with phone alignment, on the same training data. Several network architectures were investigated, and evaluated with Equal Error Rate (EER) rather than only with accuracies reported in previous works. EER represents a performance point where the true positive rate (sensitivity) equals the true negative rate (specificity). The rest of the paper is organized as follows: Section 2 overviews the proposed CTC nasal detector, Section 3 intro-

duces the experimental setup used for evaluation, and Section 4 presents results on various alternatives of the proposed detector described in Section 2. Section 5 summarises and concludes the paper.

2. NASAL SOUND DETECTION

2.1. Connectionist Temporal Classification

The connectionist temporal classification (CTC) approach [8] is an objective function that allows an end-to-end training without requiring any frame-level alignment between the input and target labels. CTC allows repetitions of output labels and extends the set of target labels with an additional *blank* symbol, which represents the probability of not emitting any label at a particular time step. It introduces an intermediate representation called the CTC *path*. A CTC path is a sequence of labels at the frame level, allowing repetitions and the blank to be inserted between labels. The label sequence can be represented by a set of all the possible CTC paths that are mapped to it.

For an input sequence $\mathbf{X} = (\mathbf{x}_1, ..., \mathbf{x}_T)$, the conditional probability $P(\mathbf{y}|\mathbf{X})$ is then obtained by summing over all the probabilities of all the paths that corresponding to the target label sequence $\mathbf{y}$ after inserting the repetitions of labels and the blank tokens, i.e.,

$$P(\mathbf{y}|\mathbf{X}) = \sum_{\hat{\mathbf{y}} \in \Omega(\mathbf{y})} P(\hat{\mathbf{y}}|\mathbf{X}) = \sum_{\hat{\mathbf{y}} \in \Omega(\mathbf{y})} \prod_{t=1}^{T} P(\hat{y}_t|\mathbf{x}_t) \quad (1)$$

where $\Omega(\mathbf{y})$ denotes the set of all possible paths that correspond to $\mathbf{y}$ after repetitions of labels and insertions of the blank token. The conditional probability of the labels at each time step, $P(\hat{y}_t|\mathbf{x}_t)$, is estimated using a neural network. The model can be trained to maximize Equation 1 by using gradient descent, where the required gradients can be computed using the forward-backward algorithm [8].

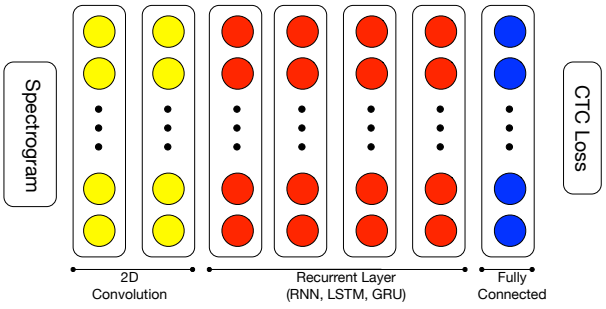

Fig. 1. *Architecture of the proposed system.*

2.2. Model Architecture

The architecture proposed in this paper is based on Deep Speech 2 model [13]. As is shown in Figure 1, the bottom of the network is two layers of convolutions over both time and frequency domains. Temporal convolution is commonly used in speech processing to efficiently model temporal invariance for variable length utterances. Convolution in frequency attempts to model spectral variance due to speaker variability and it has been shown to further improve the performance [14].

Following the convolutional layers are bidirectional recurrent layers. Different recurrent variants are explored. For recurrent layer l, the output activation $\mathbf{h}^l$ can be the standard recurrent operation

$$\mathbf{h}_t^l = \varphi(\mathbf{W}^l \mathbf{h}_t^{l-1} + \mathbf{U}^l \mathbf{h}_{t-1}^l + \mathbf{b}^l) \quad (2)$$

where $\mathbf{W}^l$ is the input-hidden weight matrix, $\mathbf{U}^l$ is the recurrent weight matrix, $\mathbf{b}^l$ is the bias term and φ is the nonlinear function. The recurrence can also be achieved through more complex recurrent units such as the Long Short-Term Memory (LSTM) units [15] and the Gated Recurrent Units (GRU) [16]. Current research in speech and language processing has shown that having a more complex recurrence can allow the network to model longer temproal dependencies between the inputs. Though many other variations exist, a recent comprehensive study showed that a GRU is comparable to an LSTM with a properly initialized forget gate bias [17], and GRU contains less parameters and is faster to train. Thus, standard RNN, GRU and LSTM are investigated as 3 variants of the nasal detection systems.

After the bidirectional recurrent layers, a fully connected layer is applied and the output is produced through a softmax function computing a probability distribution over the target labels $\{nasal, nonasal, space, blank\}$. The model is trained using the CTC loss function. To accelerate the training procedure, Batch Normalization [18] is applied on hidden layers.

3. EXPERIMENTS

3.1. Data

We used the LibriSpeech database [19] for training of the baseline and the CTC systems. The data are sampled at 16 kHz, and the training part of the corpus is split into three subsets, with size 100.6, 363.6 and 496.7 hours respectively. In addition, in this work we used the following sub-sets:

- `dev-clean` as the development data, 5.4 hours,
- `test-clean` as the evaluation data, 5.4 hours.

The training sets contain about 25 minutes of recordings per speaker, and this is limited in the gender balanced dev and test sets to about 8 minutes.

3.2. Training

3.2.1. The baseline system

The baseline system is the analysis part of the phonological vocoding described by [4], however trained on the LibriSpeech data. The training of the nasal detector requires decent phone alignment. The alignment was performed by speaker-adapted GMM model with fMLLR transforms estimated at the speaker level, trained on the whole training data (960 hours), done by the training scripts provided in [19].

The second step, training of the nasal DNN detector, starts with frame mapping the phonemes from the alignment to the nasal attribute. There are three nasal phones in English, voiced bilabial nasal [m], voiced alveolar nasal [n], and voiced velar nasal [ŋ]. The nasal DNN detector has two output labels, the particular nasal attribute occurs for the aligned phoneme or not. The training was initialised by Deep Belief Network pre-training by contrastive divergence with 1 sampling step (CD1) [20]. The 4×1024 DNNs with the softmax output function were then trained using a mini-batch based stochastic gradient descent algorithm with the cross-entropy cost function. The training was performed on the `train-clean-100` training set, and cross-validation `dev-clean` set, using the Kaldi ASR framework [21]. The training accuracy of nasal detector was 98.6% on the training data, and 98.0% on the cross-validation data.

3.2.2. The CTC system

The CTC system is based on the Deep Speech 2 model [13], and its open-source implementation[1]. The proposed CTC nasal detector is a sequence-to-sequence system. Its model starts with two layers of 2D convolutions over both time and frequency domains with 32 channels, $41 \times 11, 21 \times 11$ filter dimensions, and $2 \times 2, 2 \times 1$ stride. Four next bidirectional recurrent layers with 400 hidden units are followed by one fully connected linear layer with 4 softmax outputs $\{nasal, nonasal, space, blank\}$. The RNN models have around 3 millions (M) of parameters, GRU and LSTM models have more parameters, 8.6M and 11.4M, respectively. The input sequence are values of spectrogram slices, 20 ms long, computed from Hamming windows with 10 ms frame shifts. The input sequences are thus the values of the natural logarithm of one plus the magnitude components of the short-time Fourier transform of the windowed input signal.

The word transcription of the input signal was used to prepare the output sequences. The output (target) sequence was obtained directly from the letters of the word transcription. As all the nasal sounds in our data [m,n,ŋ] can be produced by pronouncing only the letters M and N, the output sequence was prepared without any lexicon, where all the letters M and N are converted into the $nasal$ labels, all the other letters into the $nonasal$ labels, and their repetitions were replaced by the single label. The $space$ denoted the word boundary.

The variants of the CTC nasal systems, as described in Section 2, are trained on the same training data as the baseline DNN nasal detector. We used 35 epochs to train all the models used for further evaluation.

4. RESULTS

We evaluated both the baseline DNN and CTC detectors on the same `test-clean` data set. We force-aligned the test set with the acoustic models trained on whole training LibriSpeech data (approximately 960 hours), similarly as done for the DNN baseline alignment.

We computed then the phone-attribute posterior probabilities by running forward pass of the both detectors, and counted hits of nasal and non-nasal segments. A single occurrence of the higher probability than a threshold within the phone segment was consider as a hit. The thresholds were varied to obtain the EER of the true positive and the true negative hits. While for the DNN detector the EER threshold was about 0.65, the thresholds with the CTC detectors were in range of 0.1–0.3. This is discussed later in Section 4.1.

Table 1 shows obtained EERs of the DNN and CTC nasal detectors. The CTC detectors with classic RNN and more complex LSTM achieved worse performance than the DNN detector, whereas the CTC detector with GRU outperforms the baseline.

Table 1. *Equal Error Rates (EERs) of the baseline and the proposed detection systems trained on the same training data (100 hours).*

System	EER (%)	Average CTC loss
Baseline (DNN)	4.7	–
CNN+RNN+CTC	10.2	23.307
CNN+LSTM+CTC	7.5	0.028
CNN+GRU+CTC	**4.4**	0.035

In many tasks both (GRU/LSTM) architectures yield comparable performance[2]. GRUs have fewer parameters and thus may train a bit faster or need less data to generalize. On the other hand, more data might yield greater expressive power of LSTMs, which can lead to better results.

Thus, we performed the second set of experiments with joined sets `train-clean-100` and `train-clean-360`. Table 2 shows obtained EERs for the same three CTC detector variants. Comparing to the results shown in Table 1, the LSTM+CTC system indeed yields better with more data, whereas the performance of the GRU+CTC system was saturated. Increasing the training data about 4-5 times resulted in decreasing the average CTC loss in about the same order.

[1] `https://github.com/SeanNaren/deepspeech.pytorch`

[2] `http://www.wildml.com/2015/10/`

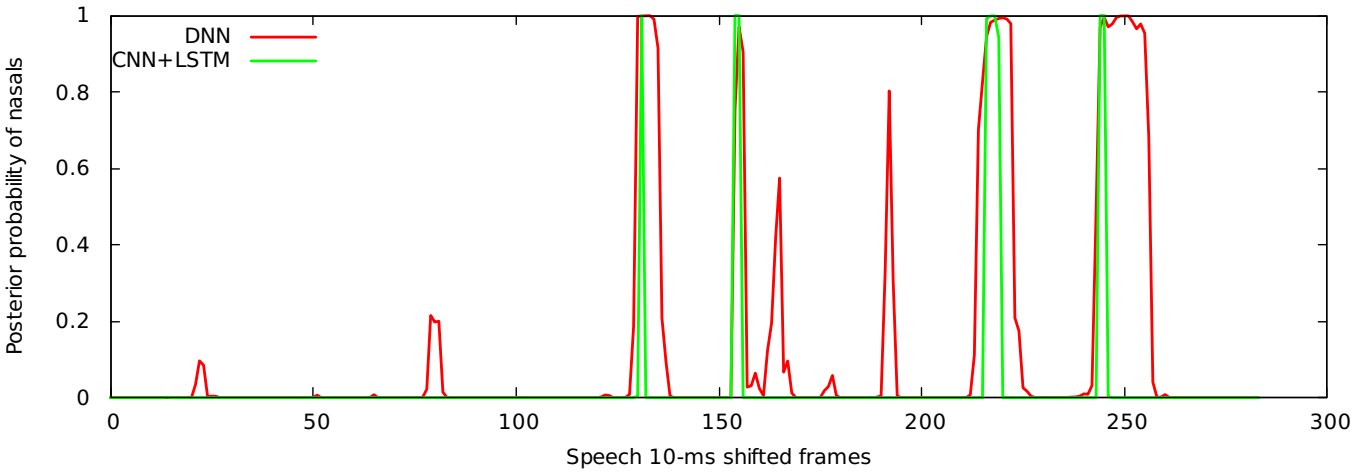

Fig. 2. *A comparison of DNN and CNN+LSTM+CTC nasal detectors for the sentence "THIS LIBRIVOX RECORDING IS IN A PUBLIC DOMAIN". To train a frame aligned CNN+LSTM+CTC system, a stride of value 1 for time dimension was used in the first CNN layer. Both detectors correctly detect 4 nasal occurrences. The CTC detector is more spiky and less noisy (the output contains less false alarms).*

Table 2. *Equal Error Rates (EERs) of the the proposed detection systems trained on bigger data (460 hours).*

System	EER (%)	Average CTC loss
CNN+RNN+CTC	4.8	6.335
CNN+GRU+CTC	4.5	0.008
CNN+LSTM+CTC	**4.1**	0.004

4.1. Discussion

The CNN in the used model performed 2D convolution, where the first dimension is frequency and the second dimension is time. A longer stride is usually applied to speed-up training. Using the stride in the time dimension results into time compacting of the input audio, e.g., using the stride of 2 results into 2 times less frames of the output. For applications where time alignment is required, we experimented with the stride of 1. The training takes twice longer as with the stride 2, but for this phone-attribute task the training still converges well.

Figure 2 shows an example of the DNN and LSTM, using stride 1 in the time domain, nasal detectors for a sample librivox recording. Firstly, the CTC output is peaky, which suggests that the CTC method can better deal with nasality characteristics due to coarticularion and better separate them from nasality associated to the features of the phoneme, than the DNN based method. The peaks slightly change the position with individual training epochs. Secondly, the CTC output contains less false positives that propagates in overall better performance, comparing to the DNN detector, in the terms of EERs. We speculate that these CTC nasal detector properties cause the lower detection thresholds observed during EER evaluation.

On the other hand, there are also some disadvantages of the proposed solution. The CTC detector does not allow straightforward application in phonological vocoding [4]. The synthesis part of the phonological vocoder converts the posterior phone attribute probabilities back to the speech samples. The peaky nature of the CTC detector breaks frame alignment of the detected phone attributes and the speech frames. One solution would be to use duration of the phone attributes at the input as well.

5. CONCLUSION

This paper has proposed to use the connectionist temporal classification for the end-to-end phone attribute modeling. The proposed system with GRUs outperformed the DNN detector trained on the same 100 hours of speech data. By increasing the training data to about 460 hours, the detector with the LSTM units, which has more parameters, outperformed the GRU detector. Comparing to the DNN detector, the CTC output is more peaky and has less false alarms.

Experiments were performed for the nasal phone attribute, focusing on selection of right architecture and evaluation. Experiments on different phone attributes is left to future work. Application of the proposed nasals detector for a phenomenon such as hypernasality, where too much air escapes through the nose while talking for example of the children with cleft palate or cerebral palsy, is also planned for future work.

The proposed system is implemented as a Python package based on PyTorch (`http://pytorch.org`). The nasal CTC detector with pre-trained models is released as the open-source code at `https://doi.org/10.24433/CO.44c5e908-1780-46cd-a950-1a2ad6ab4168`.

6. REFERENCES

[1] Ramya Rasipuram and Mathew Magimai-Doss, "Articulatory feature based continuous speech recognition using probabilistic lexical modeling," *Computer Speech & Language*, vol. 36, pp. 233–259, 2016.

[2] David J Zajac, Robert Mayo, and Ryuta Kataoka, "Nasal coarticulation in normal speakers: A reexamination of the effects of gender," *Journal of Speech, Language, and Hearing Research*, vol. 41, no. 3, pp. 503–510, 1998.

[3] Dong Yu, Sabato Siniscalchi, Li Deng, and Chin-Hui Lee, "Boosting attribute and phone estimation accuracies with deep neural networks for detection-based speech recognition," in *Proc. of ICASSP*. March 2012, IEEE SPS.

[4] Milos Cernak, Blaise Potard, and Philip N. Garner, "Phonological vocoding using artificial neural networks," in *Proc. of ICASSP*. Apr. 2015, pp. 4844–4848, IEEE.

[5] Paul Mermelstein, "On detecting nasals in continuous speech," *J. Acoust. Soc. Am.*, vol. 61, no. 2, pp. 581–587, 1977.

[6] J Glass and V Zue, "Detection and recognition of nasal consonants in american english," in *Proc. of ICASSP*. IEEE, 1986, vol. 11, pp. 2767–2770.

[7] Tarun Pruthi and Carol Y Espy-Wilson, "Acoustic parameters for the automatic detection of vowel nasalization," in *Proc. of Interspeech*, 2007, pp. 1925–1928.

[8] Alex Graves, Santiago Fernández, Faustino Gomez, and Jürgen Schmidhuber, "Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks," in *Proceedings of the 23rd international conference on Machine learning*. ACM, 2006, pp. 369–376.

[9] Dzmitry Bahdanau, Jan Chorowski, Dmitriy Serdyuk, Philemon Brakel, and Yoshua Bengio, "End-to-end attention-based large vocabulary speech recognition," in *Proc. of ICASSP*. IEEE, 2016, pp. 4945–4949.

[10] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio, "Neural machine translation by jointly learning to align and translate," *arXiv preprint arXiv:1409.0473*, 2014.

[11] Arthur S House and Kenneth N Stevens, "Analog studies of the nasalization of vowels," *Journal of Speech and Hearing Disorders*, vol. 21, no. 2, pp. 218–232, 1956.

[12] Tarun Pruthi and Carol Y Espy-Wilson, "Acoustic parameters for automatic detection of nasal manner," *Speech Communication*, vol. 43, no. 3, pp. 225–239, 2004.

[13] Dario Amodei, Sundaram Ananthanarayanan, Rishita Anubhai, Jingliang Bai, Eric Battenberg, Carl Case, Jared Casper, Bryan Catanzaro, Qiang Cheng, Guoliang Chen, et al., "Deep speech 2: End-to-end speech recognition in english and mandarin," in *International Conference on Machine Learning*, 2016, pp. 173–182.

[14] Tara N Sainath, Abdel-rahman Mohamed, Brian Kingsbury, and Bhuvana Ramabhadran, "Deep convolutional neural networks for lvcsr," in *Proc. of ICASSP*. IEEE, 2013, pp. 8614–8618.

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

[16] Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio, "Learning phrase representations using rnn encoder-decoder for statistical machine translation," in *EMNLP*, 2014.

[17] Rafal Jozefowicz, Wojciech Zaremba, and Ilya Sutskever, "An empirical exploration of recurrent network architectures," in *International Conference on Machine Learning*, 2015, pp. 2342–2350.

[18] Sergey Ioffe and Christian Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *International Conference on Machine Learning*, 2015, pp. 448–456.

[19] V. Panayotov, Guoguo Chen, D. Povey, and S. Khudanpur, "Librispeech: An ASR corpus based on public domain audio books," in *Proc. of ICASSP*. Apr. 2015, pp. 5206–5210, IEEE.

[20] Geoffrey E. Hinton, Simon Osindero, and Yee W. Teh, "A Fast Learning Algorithm for Deep Belief Nets," *Neural Comput.*, vol. 18, no. 7, pp. 1527–1554, July 2006.

[21] Daniel Povey, Arnab Ghoshal, Gilles Boulianne, Lukas Burget, Ondrej Glembek, Nagendra Goel, Mirko Hannemann, Petr Motlicek, Yanmin Qian, Petr Schwarz, Jan Silovsky, Georg Stemmer, and Karel Vesely, "The kaldi speech recognition toolkit," in *Proc. of ASRU*. Dec. 2011, IEEE SPS, IEEE Catalog No.: CFP11SRW-USB.