USING SPEECH/NON-SPEECH DETECTION TO BIAS RECOGNITION SEARCH ON NOISY DATA

Françoise Beaufays, Daniel Boies, Mitch Weintraub, Qifeng Zhu[*]

Nuance Communications
1380 Willow Road
Menlo Park, CA 94025
{francoise,boies,mw}@nuance.com

ABSTRACT

This paper focuses on the recognition of noisy speech. We show that the decoding of a noisy speech waveform can be facilitated if the recognizer has explicit knowledge of where it should hypothesize speech phones, and where it should map the acoustics to non-speech phones.

We build a speech/non-speech detector and use its output as an additional front-end feature. We show that by appropriately weighting the contribution of this feature in the decoder and by modifying the acoustic models accordingly, we can penalize speech/non-speech confusions and consequently reduce the recognition error rate.

This approach gives a 12% overall error rate reduction on a wide variety of recognition tasks and noise characteristics without degrading performance on clean test data. A simple extension of the approach boosts recognition improvements on noisy test sets to 14% overall.

1. INTRODUCTION

Recognizing speech in the presence of background noise is a notoriously difficult problem for which many approaches have been proposed in the literature, see *e.g.* [1]. In this paper, we show that a surprisingly high percentage of frames aligned to an incorrect model by the recognizer are speech frames mapped to non-speech models or vice-versa, as opposed to speech frames being mapped to the wrong speech models. This observation prompts us to build an accurate speech/non-speech (SpNsp) detector, and to use its output to modify the recognition log-likelihood function in such a way as to penalize SpNsp confusions by the decoder.

This approach is to be contrasted with recent work performed in the Aurora community, where several groups used a SpNsp detector ("voice activity detector") to label frames as speech or non-speech, and drop the non-speech frames during recognition (see *e.g.* [2, 3, 4]). This has the effect of eliminating speech insertion errors in long trailing silence and noise segments, and of significantly reducing error rates on the Aurora databases.

The test data considered in this work is tightly endpointed, so we don't expect frame dropping to help any further. However, in spite of its endpointing, the data still contains non-speech segments, for example all the short pauses that occur between words. For those frames, we influence the recognizer to correctly hypothesize non-speech phones. More importantly, our approach also tries to prevent the recognizer from hypothesizing non-speech phones during segments labeled as speech by the detector, which frame dropping cannot achieve.

This work also relates to a body of literature concerned with the developement of linguistically motivated front-ends (see *e.g.* [5, 6, 7]). In these studies, a probability of speech is often part of the proposed front-end. Our approach differs from these in its use of the SpNsp detector to directly affect the recognition search.

The SpNsp detector developed in this work consists of a neural network whose inputs are a series of knowledge-based features targetted at the identification of specific speech classes (*e.g.* voiced sounds, nasals). These features are combined in a data-driven fashion to estimate the probability that the current frame is speech. This probability estimate (as opposed to a 0/1 decision in frame dropping) is then used to softly penalize the decoder for misaligning frames to the wrong phone classes.

2. WHY FOCUS ON SPEECH/NON-SPEECH?

In an error-analysis experiment, we considered a set of waveforms with their recognition hypotheses (Hyp), word transcriptions (Ref), and the phone-level segmentations of both. We collected statistics on the frame-level errors, *i.e.* frames where the Hyp and Ref phones differ. The relative importance of each type of error is summarized in Table 1.

Nsp → Sp	Sp → Nsp	Nsp → Nsp	Sp → Sp
17%	47%	5%	31%

Table 1. Distribution of Frame Error Types in a Typical Recognition Run.

Table 1 shows that

1. More than half the frame-level errors are SpnSp confusions (17 + 47 = 64% of all frame errors)
2. Speech/Speech confusion does *not* dominate the errors as we might have expected (31% of all frame errors).

Of course, not all frame errors correspond to recognition errors, but a majority do, and this indicates that SpNsp distinctions are not well covered by the baseline recognition system.

[*] Now at ICSI, Berkeley, CA.

3. SYSTEM ARCHITECTURE

Figure 1 illustrates the proposed architecture. A SpNsp detector estimates the probability that the current frame is speech. This probability is scaled by a constant d (we'll discuss the purpose of this constant shortly), and appended to the baseline front-end feature vector.

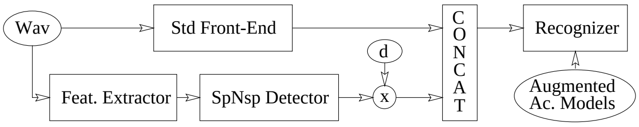

Fig. 1. Architecture of the Proposed System.

The augmented feature vector is then fed into a standard 3-state triphone hidden Markov models recognizer, with a Genone-based state clustering mechanism [8], and diagonal covariance matrices.

3.1. Modification of the Acoustic Models

Clearly, the acoustic models used for recognition must be modified to account for the additional SpNsp feature. A straightforward solution would be to retrain the models with the augmented front-end. Instead, we chose to modify existing N-dimensional models by artificially adding an $(N+1)^{th}$ dimension to all the Gaussians in the models.

We realize this by tying along the $(N+1)^{th}$ dimension all the Gaussians that model speech phones, and all the Gaussians that model non-speech phones. The non-speech Gaussians's parameters are set to $(\mu_{Nsp}, \sigma^2_{Nsp}) = (0, 1)$, and the speech Gaussians's parameters are set to $(\mu_{Sp}, \sigma^2_{Sp}) = (d, 1)$, *i.e.* the speech and non-speech models are placed at a distance d from each other along the new dimension.

The use of fixed "binary" models for the $(N+1)^{th}$ feature is motivated by two reasons. First, it simplifies our experimentation with the SpNsp detector: we can modify it any time without having to retrain the acoustic models. Second, it is compatible with the discriminative training of the detector (see Section 4.2) whose output is a posterior probability comprised between 0 and 1 (or 0 and d after rescaling).

3.2. Effect of the Additional Feature on the Log-Likelihood Function

The adjunction of an $(N+1)^{th}$ feature in the front-end modifies the decoding log-likelihood function according to

$$\text{LogLik}(N+1) = \text{LogLik}(N) + AdditionalCost \qquad (1)$$

Assuming a perfect SpNsp detector whose output, P_{Sp}, is 0 or 1 (or 0/d after scaling), a speech frame scored against a speech model has an *AdditionalCost* equal to $(P_{Sp} - \mu_{Sp})^2/\sigma^2_{Sp} = (d - d)^2/1 = 0$. The same speech frame scored against a non-speech model has an *AdditionalCost* of $(P_{Sp} - \mu_{Nsp})^2/\sigma^2_{Nsp} = (d - 0)^2/1 = d^2$. Likewise a non-speech frame has a cost of 0 when scored against a non-speech model, and a cost of d^2 when scored against a speech model. The adjunction of the *AdditionalCost* effectively penalizes SpNsp confusions during the decoding process.

The value of d can be optimized to balance the contribution of SpNsp to other phone confusions: with $d = 0$, there is no SpNsp penalty, but as d increases, SpNsp errors start dominating the overall cost function. By performing a sweep over d and tracking the recognition performance, one can choose the optimal weight, d, for a given SpNsp detector, again without any model retraining.

This explicit scaling of the SpNsp feature prevents it from being overwhelmed by the other N features in the log-likelihood function, a property that is not shared by a Karhunen-Loeve decorrelation of the feature vector [6, 7].

4. SPEECH/NON-SPEECH DETECTION

The previous section described how a SpNsp detector can be used to impose a soft SpNsp segmentation of the waveform on the decoder. This section instead describes how we build the detector. The general approach is that of defining knowledge-based features, and assembling them in an automatic data-driven fashion. For example, we derive features targetted at the identification of voiced or fricative sounds, but rather than explicitely trying to mark each frame as voiced or fricative (and therefore speech as opposed to non-speech), we combine these features with a neural network which we optimize to distinguish between speech and non-speech, ignoring the original intended purpose of each feature. This gives more flexibility to the feature combiner to use multiple cues to derive its final estimate.

4.1. Speech/Non-Speech Features

The SpNsp features used in this work include:

- *Distance to Voicing*:

  A standard pitch tracker estimates the voicing level profile of the waveform. Regions above a given threshold are marked as voiced. The distance to voicing is defined as the distance between the current frame and the closest voiced frame. A distance of zero indicates that the frame is voiced, and thus speech. A large distance hints that the frame is probably non-speech since human speech typically doesn't contain long segments with no voicing.

- *Frame Energy*:

  The energy of a frame is a rough indicator of its SpNsp status (waveforms are amplitude-normalized prior to feature extraction).

- *Voicing Level*

  This helps identifying as speech voiced frames whose voicing level may have been too low to exceed the voicing threshold.

- *Spectral Tilt*

  The spectral tilt is defined as the ratio of high- to low-frequency energies. Fricatives typically display a larger spectral tilt than steady-state noises such as car noise.

- Various combinations of the above features and their referencing w.r.t. the background noise level.

4.2. Neural Network Feature Combiner

For each frame of data, the above features are evaluated and inputted in a 3-layer feed-forward neural network with 400 hidden nodes. The neural network is trained on a database of 13K sentences that have no overlap with the test data, but that display roughly the same distribution of acoustic characteristics (noise types and levels, communication channels, etc). The network is trained to minimize the cross-entropy between its outputs and the known status of the training frames: speech or non-speech.

5. RECOGNITION EXPERIMENTS

The baseline system used in these experiments is a general purpose triphone-state HMM recognizer, with a 27-dimensional melfilterbank cepstral coefficient front-end, cepstral mean subtraction and standard noise reduction. The baseline acoustic models are trained with a large amount of phonetically rich, acoustically varied, telephone speech.

We use a wide variety of test sets to exemplify different acoustic conditions, grammars, and applications. For conciseness, we group these into four databases as described below.

- *CarNoisy*
  This database contains speech collected handsfree in a car, over the cellular network. It is generally very noisy. It contains about 4K utterances from 300 speakers, and spans 2 grammars: universal commands and digit or alphadigit ID numbers.
- *CarNoiseSup*
  This database also consists of handsfree in the car cellular speech, but an aggressive noise suppression processing has been applied by the handsfree kit. As a result, there is little perceivable background noise, but the speech segments remain noisy. The data also displays short, loud, mechanical noises. The database contains 9K utterances from 100 speakers, and combines 3 grammars: name and number dialing, stock quotes, and travel arrangements with dates and city/state destinations.
- *CarMultiMic*
  This data was collected handsfree in the car, with different microphones and microphone placements. The background noise varies in level and is essentially steady. The database contains 15K utterances. The grammar allows for name and number dialing, and traffic and weather report querries.
- *CleanHandH*
  This database contains a collection of clean landline and cellular telephone test sets assembled from a large number of speakers. It combines many test sets spanning a variety of small and large vocabulary grammars. It was used to control the performance of the approach on data that is not affected by noise.

All the error rates reported below are natural language error rates (NLERR), and are thus measured at the string level as opposed to the word level.

5.1. Perfect Knowledge Experiments

The experiments reported in this section aim at answering the question: “What recognition gain can we expect with the proposed approach, assuming we had a perfect SpNsp detector?”.

To answer this question, we performed a set of recognition experiments where the SpNsp feature was obtained for each frame of test data by looking up the forced alignment of the waveform to its reference word string, and defining the SpNsp feature as 0 if the frame was aligned to a non-speech phone, and d otherwise. The experiment was repeated for different values of d. The average recognition error rate over the three noisy test databases is shown in Fig. 2.

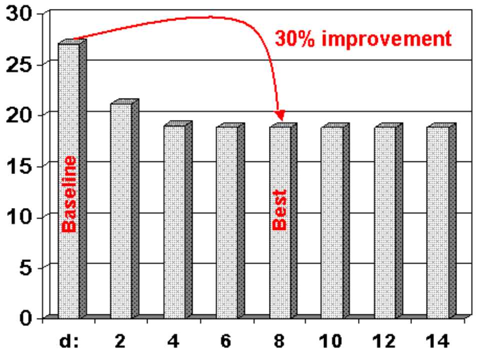

Fig. 2. Average NLERR over all the noisy test sets as a function of the model distance d, assuming perfect SpNsp knowledge.

Fig. 2 shows that the error rate significantly decreases as the distance between the speech and non-speech models, d, increases (*i.e.* a higher penalty is imposed on the decoder for SpNsp confusions), up to an optimal value at $d = 8$. Passed that value, the error rate starts increasing slowly, probably because of search errors. The best NLERR improvement is roughly 30%.

Table 2 shows the breakdown of error rates for the different databases. The improvements are roughly similar for all conditions, including the clean test sets.

	Baseline	w/ Perfect SpNsp	Rel. NLERR Improv.
CarNoisy	18.3	10.2	44
CarNoiseSup	25.3	17.2	32
CarMultiMic	30.1	21.7	28
CleanHandH	9.4	7.1	26

Table 2. NLERR with the baseline and SpNsp augmented recognition systems, assuming perfect SpNsp detection.

Though these numbers are encouraging, it should be anticipated that the gains reported in Table 2 are significantly higher than what we may obtain with a real SpNsp detector: the detector will not be perfect, and its output will take the form of a probability to model the uncertainty we have in the SpNsp estimate. Moreover, the “truth” in these experiments comes from forced alignments, which essentially “do what the recognizer wants to see”, *i.e.* match its modeling and alignment idiosyncrasies.

5.2. Recognition Experiments with the SpNsp Detector

This section summarizes the results we obtained with the neural net SpNsp detector.

Figure 3 shows the average error rate over the noisy test sets, as a function of the free parameter, d. The overall NLERR improvement is 12%, compared to 30% in the perfect SpNsp experiments.

Also, the optimal value of d decreased from 8 to 2.5, indicating that the recognizer can trust the neural net detector significantly less that the perfect detector.

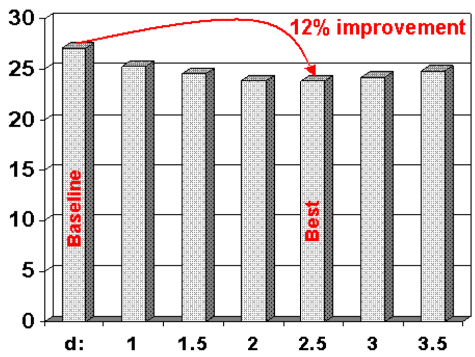

Fig. 3. Average NLERR over all the noisy test sets as a function of the model distance d, using the neural net SpNsp detector.

Results for individual databases are summarized in Table 3. The NLERR gains are uniform under the different noisy conditions, but we observe no improvement on the clean data. This is somewhat unexpected given the significant gain on this data with the perfect SpNsp detection assumption. A possible explanation is that in order to improve clean performance with this approach, one would need a tighter SpNsp detection than what is necessary to improve on the noisy conditions, and that we haven't reached that level of accuracy with our current SpNsp detector.

	Baseline	w/ NNet SpNsp	Rel. NLERR Improv.
CarNoisy	18.3	15.8	14
CarNoiseSup	25.3	22.0	13
CarMultiMic	30.1	26.8	11
CleanHandH	9.4	9.3	2

Table 3. NLERR with the baseline and SpNsp augmented recognition systems.

5.3. Recognition Experiments with SpNsp and Voicing Detectors

We saw in Section 4.1 that a major component of the SpNsp dectector is a voicing detector. Since this detector is available, we can also use it directly to provide an $(N+2)^{th}$ feature. Again, the acoustic models are augmented in a manner similar to that described in Section 3.1, setting the model means to 0 for unvoiced phones, and to d' for voiced phones. Recognition experiments with both features showed that a distance $d' = 2$ was an appropriate choice. Results with both features are summarized in Table 4. The overall NLERR improvement on noisy test sets increases from 12 to 14%, improvements on the clean test sets remain negligeable.

	Baseline	w/ SpNsp + Voicing	Rel. NLERR Improv.
CarNoisy	18.3	15.0	18
CarNoiseSup	25.3	21.2	16
CarMultiMic	30.1	26.3	13
CleanHandH	9.4	9.2	3

Table 4. NLERR with the baseline and (SpNsp + Voicing) augmented recognition systems.

6. CONCLUSIONS

In this paper, we proposed to use a SpNsp detector to impose a soft SpNsp segmentation of the waveform to the decoder, thereby easing its task of "filling in" the speech segments. We showed that:

1. The SpNsp detector output can be used as an additional front-end feature, provided that it is scaled appropriately to control the impact of the additional feature on the search.
2. The baseline acoustic models can easily be modified to account for the new feature while conveniently avoiding any model retraining during the SpNsp detector optimization.
3. A reasonably accurate SpNsp detector can be obtained by combining acoustic features aimed at the identification of specific linguistic classes into a neural network that is optimized for SpNsp detection.

This approach brings a consistent 12% relative NLERR improvement w.r.t. a state-of-the-art recognition baseline system over a wide variety of stationary and non-stationary background noises and recognition tasks. The NLERR gain can be boosted to 14% by adding in voicing as a second additional feature.

7. REFERENCES

[1] Y. Gong, "Speech Recognition in Noisy Environments: A Survey", *Speech Communication*, 16, pp. 261-291, 1995.

[2] Y.M. Cheng, D. Macho, Y. Wei, D. Ealey, H. Kelleher, D. Pearce, W. Kushner, T. Ramabadran, "A Robust Front-end for Distributed Speech Recognition", *Proc. Eurospeech'01*.

[3] B. Andrassy, D. Vlaj, Ch. Beaugeant, "Recognition Performance of the Siemens Front-end with and without Frame Dropping on the Aurora 2 Database", *Proc. Eurospeech'01*.

[4] C. Benítez, L. Burger, B. Chen, S. Dupont, H. Garudadri, H. Hermansky, P. Jain, S. Kajarekar, N. Morgan, S. Silvadas, "Robust ASR Front-End Using Spectral-Based and Discriminant Features: Experiments on the Aurora Tasks", *Proc. Eurospeech'01*.

[5] K. Kirchhoff, "Combining Articulatory and Acoustic Information for Speech Recognition in Noisy and Reverberant Environments", *Proc. ICSLP'98*.

[6] B. Launay, O. Siohan, A. Surendran, C.H. Lee, "Towards Knowledge-Based Features for HMM Based Large Vocabulary Automatic Speech Recognition", *Proc. ICASSP'02*.

[7] S. Sivadas, and H. Hermansky, "Hierarchical Tandem Feature Extraction", *Proc. ICASSP'02*.

[8] V. Digalakis, P. Monaco, and H. Murveit, "Genones: Generalized Mixture Tying in Continuous Hidden Markov Model-Based Speech Recognition", *IEEE Trans. on Speech and Audio Processing*, pp. 281-289, 1996.