# A dynamic programming framework for neural network-based automatic speech segmentation

[1]*Van Zyl van Vuuren,* [2]*Louis ten Bosch,* [1]*Thomas Niesler*

[1]Department of Electrical and Electronic Engineering, University of Stellenbosch, South Africa
[2] Department of Linguistics, Radboud University, Nijmegen, The Netherlands
{vzvv,trn}@sun.ac.za, l.tenbosch@let.ru.nl

## Abstract

Neural networks have recently been shown to be a very effective approach to the unconstrained segmentation of speech into phoneme-like units. The neural network is trained to indicate when a short local sequence of feature vectors is associated with a segment boundary, and when it is not. Although this approach delivers state-of-the-art performance, it is prone to over-segmentation at ambiguous segment boundaries. To address this, we propose the incorporation of the neural network segmenter into a dynamic programming (DP) framework. We evaluate the DP-based approach on the TIMIT corpus, and show that it leads to improved performance.

**Index Terms**: unconstrained automatic speech segmentation, dynamic programming, multilayer perceptrons

## 1. Introduction

The task of accurately segmenting a speech signal into phoneme-like units plays an important role in the speech processing field. Although accurate manual segmentation can be achieved by trained phoneticians, the task is tedious, expensive and subjective. In an under-resourced setting, in which very little transcribed phonetic material is available, automatic segmentation algorithms can accelerate the task of developing a pronunciation dictionary and obtaining suitable bootstrapping acoustic training data, thereby substantially reducing the time it would take to develop an automatic speech recognition (ASR) system. The availability of reliable automatic segmentation algorithms is also useful in technologies outside ASR, such as the study of pronunciation variation, the development of coherent large-scale dictionaries, text-to-speech (TTS) applications [1], and many others [2, 3, 4].

A distinction can be made between segmentation approaches that require phone or orthographic transcripts, and those that do not. These two approaches are often referred to as constrained and unconstrained respectively [5]. There is also a distinction between algorithms that segment speech into syllables and into phoneme-like units.

Constrained approaches usually perform a forced alignment between phoneme based hidden Markov models (HMMs) and a phonetic transcription [1, 5, 6], or align phoneme templates to a signal using dynamic time warping (DTW) [1, 5, 7]. Unconstrained approaches typically rely on a scoring function that is applied at the feature level and indicates possible segment boundaries. Because these scores are calculated from a local group of features, we will refer to them as 'local scores'. Popular local scores are vector distance functions which respond to the dynamics of the features and from which a peak-picking algorithm finds viable local maxima at which to hypothesise segment boundaries [8, 9, 10, 11, 3]. Rule based approaches that use language-specific knowledge to calculate a local score independent of the phone string [4, 12], and HMM phone loop segmentation also fall into the unconstrained class [5].

Artificial neural networks (ANNs) have been applied to both constrained and unconstrained segmentation approaches. The constrained approaches are mostly based on hybrid HMM/ANN algorithms in which multilayer perceptrons (MLPs) act either as phone probability estimators [13, 14], or are used to detect phoneme transitions in order to refine the boundaries produced by an HMM alignment [15, 16]. For unconstrained segmentation, ANNs have recently been shown to be highly effective [5, 17]. We propose an improved unconstrained ANN segmentation algorithm by introducing a dynamic programming (DP) framework which employs a probabilistic segment length model in conjunction with the ANN scores to hypothesise segment boundaries. The TIMIT corpus will be used for the training and testing of the proposed algorithm.

Section 2 gives a brief overview of the ANN-based segmentation algorithm used as the baseline, Section 3 describes the DP algorithm, Section 4 discusses the evaluation methods used, Section 5 gives an overview of the experimental setup, Section 6 contains the results, and conclusions are given in Section 7.

## 2. Segmentation using neural networks

An MLP can be employed to compute a local score on the basis of a group of consecutive feature vectors. In recent work this was achieved by training two output neurons, one outputs a high value when the evidence in the input feature vectors supports the presence of a boundary, and the other when the evidence supports the absence of a boundary [5]. The training data consists of feature vector groups located around phoneme boundaries and feature vector groups midway between two boundaries. The local score is obtained by taking the difference between the two outputs. Approaches that rely on the detection of local maxima in such local scores, such as those proposed in [8, 9, 10, 11, 3], may now be employed to find possible segment boundaries.

As proposed in [5], an MLP with 30 hyperbolic tangent neurons in the hidden layer, and two hyperbolic tangent neurons in the output layer was employed for segmentation in our experiments. A window size of 10ms with a step of 5ms was used to calculate the feature vectors. Groups of 11 consecutive feature vectors centred about the point of interest were used with 12 MFCCs and log energy as features. The network was trained by back-propagation, and functions by detecting regions in time where the local score is larger than zero throughout the region. A segment boundary is then hypothesised at the frame

Copyright © 2013 ISCA

at which the local score is at a maximum within the region as demonstrated by Equation 1,

$$[\hat{B_R}] = \underset{t\epsilon\{S_R...E_R\}}{\operatorname{argmax}} \{LS(i_t)\} \tag{1}$$

where $B_R$ is the boundary frame in the region, $S_R$ and $E_R$ are the start and end of the region respectively, $LS$ is the local score, and $i_t$ are the frames between $S_R$ and $E_R$ [5].

# 3. DP-based segmentation

We propose to embed the neural-network based detection mechanism described in the previous section in a dynamic programming (DP) framework by including an explicit probabilistic model for the length of a segment. In this way segments that are either very short or very long are penalised by their associated low probability. The probability distribution of phoneme lengths for TIMIT is illustrated in Figure 1. For illustrative purposes, the distribution has been normalised with respect to its maximum.

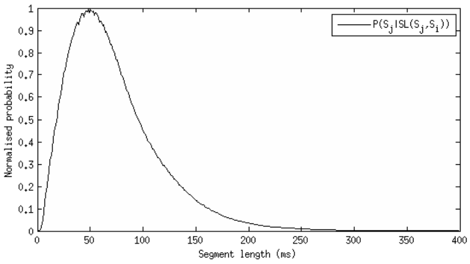

Figure 1: Probability distribution of phoneme lengths in the TIMIT training set [18], as used by Equation 2.

The DP-based segmentation algorithm will in general need to compute probabilities for segments as long as the utterance itself. This is achieved by attaching a linear decaying tail stretching from the maximum segment length for which a probability estimate is available to the length of the utterance at hand.

## 3.1. Local score probability distributions

To gain some insight into the behaviour of the local score near segment boundaries, its probability distribution in boundary regions can be estimated. A similar distribution can be determined for regions that are far from these boundaries. Figure 2 shows these distributions for the local score when calculated with the MLP proposed in Section 2. The distributions were estimated from the TIMIT core test set, and are normalised with respect to their maxima for clarity. The distributions shown in Figures 1 and 2 are used to determine the probability that a boundary occurs at a specific frame in a speech signal.

## 3.2. Dynamic programming

Consider an utterance consisting of N+1 frames. Let the time of occurrence of each frame correspond to a state of an HMM as shown in Figure 3, where M is the maximum allowed number of frames per segment and $S_0$ is the time of occurrence of the first frame of the signal. The vertical dashed arrows between $S_1$ and $S_1$, and between $S_{N-1}$ and $S_{N-1}$ indicate an expansion of the same HMM state.

When a state is visited by a path through the Markov model, a segment boundary is considered to occur at the corresponding speech frame. The transition and emission probabilities are calculated according to Equations 2 and 3 respectively, where $SL$ refers to the segment length, $LS$ to the local score, and $SB$ to the occurrence of a segment boundary.

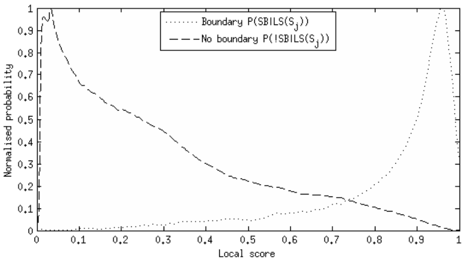

Figure 2: Estimated probability distribution of the MLP local score values at, and away from, phoneme boundaries (Eq. 3).

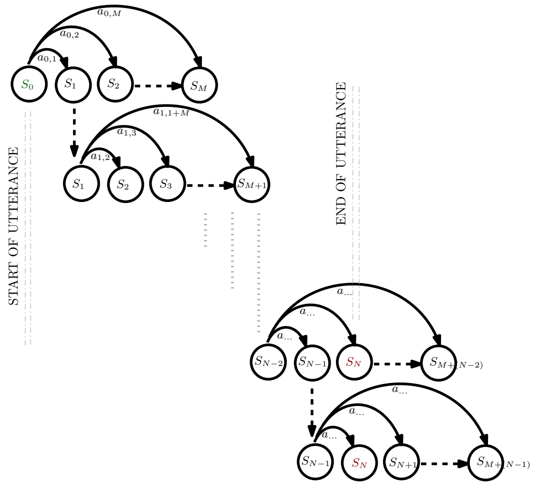

Figure 3: DP-based segmentation formulated as an HMM.

$$a_{i,j} = P(S_j|SL(S_j, S_i)) \tag{2}$$

$$b_j = P(SB|LS(S_j)) \tag{3}$$

The segment length is given by Equation 4, where $step$ is the frame step in seconds.

$$SL(S_j, S_i) = (j - i) * step \tag{4}$$

Hence the transition probability is dependent only on the elapsed time between states, while the emission probability at state $Sj$ is dependent on the local score $LS(Sj)$. The emission probability can be calculated by the application of Bayes rule as shown in Equation 5, where $!SB$ refers to the absence of a segment boundary.

$$P(SB|LS(S_j)) = \frac{P(LS(S_j)|SB)P(SB)}{P(LS(S_j)|SB)P(SB) + P(LS(S_j)|!SB)P(!SB)} \tag{5}$$

The prior probability of the occurrence of a segment boundary can be estimated from the TIMIT corpus, as shown in Equation 6.

$$P(SB) = \frac{number\ of\ phoneme\ boundaries\ in\ TIMIT}{number\ of\ frames\ in\ TIMIT} \tag{6}$$

The probability that a boundary occurs at a particular frame can now be calculated by using Equations 4 and 5 in conjunction with estimates of the probability distributions.

### 3.3. Optimal path

The globally optimal path from $S_0$ to $S_N$ in Figure 3 can be determined using the Viterbi algorithm. The states that are visited by the optimal path identify the optimal segmentation. States $S_0$ and $S_N$ are always included in the path, and therefore the algorithm assumes that segment boundaries are always present at the start and the end of the speech signal. This means that any initial and final silence must be removed before applying the algorithm.

### 3.4. Normalising for path length

During the Viterbi decoding, many probabilities are multiplied together for any given path. Shorter paths (which contain fewer multiplications and thus longer segments) may therefore be preferred, even when these have low associated emission and transition probabilities. We compensate for this by modifying the emission and transition probabilities as shown in Equations 7 and 8.

$$a_{i,j} = P(S_j|SL(S_j, S_i))^{SL(S_j,S_i)} \tag{7}$$

$$b_j = P(SB|LS(S_j))^{SL(S_j,S_i)} \tag{8}$$

This modification normalises segment probabilities with respect to their lengths.

## 4. Assessing segmentation accuracy

In order to assess the quality of automatic-generated segmentations, we will determine how closely they correspond to the TIMIT phonetic segmentations.

### 4.1. Comparing segmentations by fixed margins

It appears to be standard practice in related research to consider a hypothesised and a reference segmentation boundary to be a match whenever they occur within 20ms of one another [1, 5, 6, 8, 9, 10, 11, 14, 16]. All non-matching boundaries are then regarded as either insertions or deletions. In order to make our results comparable to those of others, this scoring framework has been employed. An error measure termed the **average error** (ERR) is calculated. This figure is the average percentage insertions (INS) and deletions (DEL) taken with respect to the number of reference boundaries in the utterance.

### 4.2. Comparing segmentations by DP

Two sequences of segment boundary times can also be compared using DP. We proceed by first determining the best alignment between the two sequences of boundary times. The total absolute time difference between the boundaries paired in this alignment is taken as the cost of the path. By dividing the path cost by the number of reference boundaries, the cost in seconds per reference boundary can be obtained. This will be referred to as “DP Cost” and used as a figure of merit in our experiments.

A disadvantage of the fixed margin method is that all insertions and deletions are considered equal regardless of their positions. For example, a succession of deletions is not explicitly penalised in the fixed margin method. Comparing segmentations by DP penalises insertions and deletions relative to their closest paired boundary, and a succession of deletions will therefore result in a large cost because the closest paired boundary will be far away. We will use the DP evaluation mechanism in addition to the fixed margin method to obtain a better impression of how closely two boundary sequences are aligned.

## 5. Experimental setup

### 5.1. Data

Our experimental evaluations are based on the TIMIT database [19]. The development set specified in [18] was used to optimise all parameters, and the core test set was reserved exclusively for final testing. There is no speaker overlap between any of the sets. Leading and trailing silences were removed to ensure that each utterance begins and ends with a segment boundary.

### 5.2. Probability weights

As it stands, the DP segmentation algorithm will give equal weight to the transition and emission probabilities, due to the segment length and local score respectively. However, it may be beneficial to shift the balance more strongly towards one or the other. By multiplying the log values of the emission and transition probabilities by positive constants that sum to one, this shift in balance can be achieved, and will allow deletions to be traded for insertions and vice versa. In all our experiments, this value was optimised on the development set.

## 6. Results

Table 1 compares the performance of the NN based segmentation algorithm proposed in [5], the same algorithm when embedded in the DP framework we propose in Section 3, and a combined approach that will be discussed in Section 6.1. Note that the results presented by [5] were calculated by using different assessment measurements to those used in our paper, and are therefore not directly comparable.

| Method | DP Cost (ms) | %INS | %DEL | %ERR |
|---|---|---|---|---|
| NN | 14.50 | 12.74 | 17.00 | 14.87 |
| DP | 13.57 | 12.46 | 17.96 | 15.21 |
| Combined | 13.20 | 13.06 | 15.72 | 14.39 |

Table 1: Comparison of core test set performance of NN based segmentation in isolation, when embedded in a DP framework, and when NN and DP approaches are combined.

In our experiments, we attempted to keep the deletion and insertion errors of the DP segmentation close to those obtained in [5] to facilitate comparison. Table 1 shows that an increase in the average error is observed for the DP segmentation method. On the other hand, a considerable decrease in the DP cost was achieved. This indicates that the DP leads to segment boundaries that are on average more closely aligned to the TIMIT boundaries, but with slightly fewer boundary pairs within 20ms of one another.

During informal evaluation we observed typical errors made by both the NN and DP algorithms. We illustrate some of these errors by means of the segmentations produced by both algorithms for the same sentence, dr6-fbch0-sa1, in Figures 4 (NN) and 5 (DP). Each figure shows the first two seconds of the utterance. The dashed vertical lines show the hypothesised boundaries, and the solid vertical lines show the TIMIT reference phone boundaries.

For the NN algorithm, insertions were found to be frequent at boundary regions where local scores had very small amplitudes. The boundary in the middle of the second ‘r’ in Figure 4 illustrates this type of insertion. The NN method also has a tendency to miss segment boundaries when more than one large peak occurs in the boundary region. The deletion of the boundary at the start of the second ‘s’ illustrates this tendency.

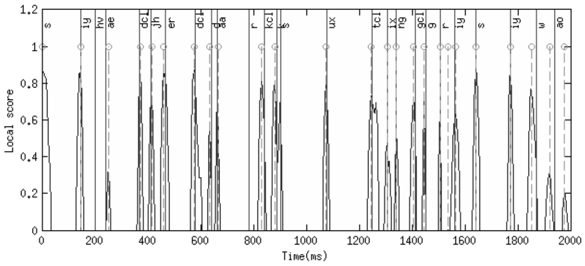

Figure 4: Segmentation results for the NN algorithm for sentence dr6-fbch0-sa1.

The DP implementation, on the other hand, frequently skips a plausible boundary because it closely precedes a boundary with relatively high probability, leading to a deletion. The deletion near ‘d’ in Figure 5 is an example of such an error.

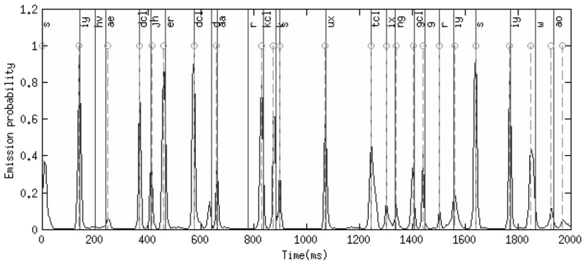

Figure 5: Segmentation results for the DP algorithm for sentence dr6-fbch0-sa1.

### 6.1. Emission probability threshold

The local scores generated by the MLP appear to be very good indicators of the presence of boundaries. The tendency of plausible boundaries to be skipped by the DP implementation is therefore detrimental to its performance when using the MLP-based local score. We attempted to address this shortcoming by proposing a hybrid approach between the two algorithms. This hybrid approach takes all the boundaries proposed by the NN segmentation approach that have an emission probability above a specific threshold, and fixes these before DP is performed. The DP then has the task of choosing between boundaries hypothesised by peaks with lower emission probabilities as well as between other higher probability peaks that were skipped by the NN segmenter. Figures 6 and 7 show the effect of this threshold on the DP cost and the average error respectively.

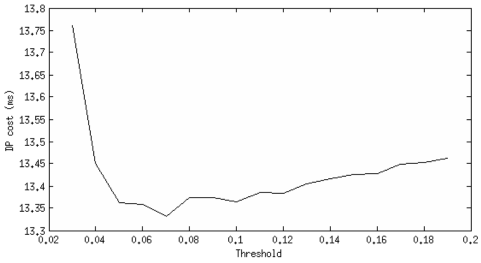

Figure 6: DP cost (as described in Section 4.2) as a function of the emission probability threshold for the combined method, measured on the development set.

Figures 6 and 7 show that segmentation accuracy is improved by including the new threshold. For the optimal threshold, Table 1 gives the segmentation performance achieved by the combined method on the core test set. There is a substantial improvement in terms of both DP cost and average error, indicating that the hypothesised boundaries are better aligned with the TIMIT boundaries and that more boundary pairs are within 20ms of one another.

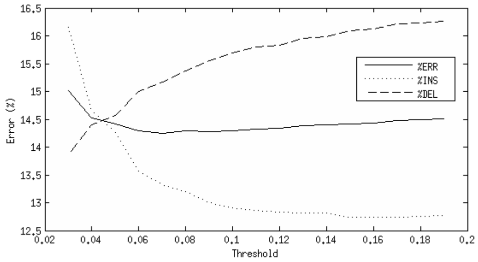

Figure 7: Average error (as described in Section 4.1) as a function of the emission probability threshold for the combined method, measured on the development set.

The segmentation of sentence dr6-fbch0-sa1 achieved by the combined method is shown in Figure 8. Both the insertion at the second ‘r’ and the deletion near the ‘d’ have been avoided. While this is merely a specific example, similar improvements were observed informally for many other utterances.

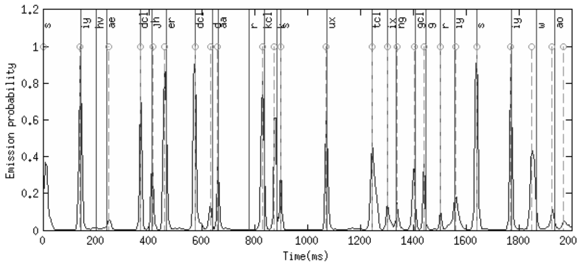

Figure 8: Segmentation results for the combined method for sentence dr6-fbch0-sa1.

## 7. Summary and conclusion

We propose an algorithm based on the principle of dynamic programming for the unconstrained automatic segmentation of continuous speech into phoneme-like units. A measure of segment boundary probability is computed by an MLP from a number of consecutive feature vectors. This is combined with a knowledge of the statistical distribution of the segment lengths within a dynamic programming framework to obtain an optimal placement of segment boundaries. We compare the performance of our algorithm with the performance of a recently proposed alternative, which applies MLPs, but not within a DP framework. For experimental evaluation, we measure how closely the hypothesised boundaries match the TIMIT phone boundaries. It was found that an improved alignment between the generated and TIMIT boundaries was achieved when employing the DP-based framework, and that a hybrid approach which combines aspects of both algorithms leads to even better results. We conclude that the incorporation of dynamic programming into speech segmentation algorithms is successful.

## 8. Acknowledgements

We gratefully acknowledge the financial assistance of the National Research Foundation (NRF) towards this research. Opinions expressed and conclusions arrived at are those of the authors and are not necessarily to be attributed to the NRF.

# 9. References

[1] J. Adell, A. Bonafonte, J. Gomez, and M. Castro, “Comparative study of automatic phone segmentation methods for TTS,” in *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing, ICASSP*, vol. 1, 2005, pp. 309 – 312.

[2] M. Sharma and R. Mammone, “‘Blind’ speech segmentation: Automatic segmentation of speech without linguistic knowledge,” in *Proceedings of the Fourth International Conference on Spoken Language Processing, ICSLP*, vol. 2, oct 1996, pp. 1237 –1240.

[3] L. ten Bosch and B. Cranen, “A computational model for unsupervised word discovery,” in *Order: A Journal On The Theory Of Ordered Sets And Its Applications*, 2007, pp. 1 – 4.

[4] D. Wang, L. Lu, and H.-J. Zhang, “Speech segmentation without speech recognition,” in *Proceedings of International Conference on Multimedia and Expo, ICME*, vol. 1, july 2003, pp. 405 – 408.

[5] V. Keri and K. Prahallad, “A comparative study of constrained and unconstrained approaches for segmentation of speech signal.” in *Proceedings of International Conference on Spoken Language Processing, ICSLP*, 2010, pp. 2238–2241.

[6] S. Hoffmann and B. Pfister, “Fully automatic segmentation for prosodic speech corpora,” in *Proceedings of the International Conference on Spoken Language Processing, ICSLP*, 2010, pp. 1389–1392.

[7] F. Malfrre and T. Dutoit, “High-quality speech synthesis for phonetic speech segmentation,” in *Fifth European Conference on Speech Communication and Technology, EUROSPEECH, Rhodes, Greece, September 22-25*, 1997.

[8] Okko Räsänen, U. K. Laine, and T. Altosaar, “Blind segmentation of speech using non-linear filtering methods,” in *Ipsic I. (Ed.): Speech Technologies*. InTech Publishing, 2011, pp. 105 –124.

[9] A. Sarkar and T. Sreenivas, “Automatic speech segmentation using average level crossing rate information,” in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing, ICASSP*, vol. 1, 2005, pp. 397 – 400.

[10] Y. P. Estevan, V. Wan, and O. Scharenborg, “Finding maximum margin segments in speech,” in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing, ICASSP*, 2007, pp. 937 –940.

[11] G. Aversano, A. Esposito, and M. Marinaro, “A new text-independent method for phoneme segmentation,” in *Proceedings of the 44th IEEE 2001 Midwest Symposium on Circuits and Systems, MWSCAS*, vol. 2, 2001, pp. 516 –519.

[12] K. Hatazaki, Y. Komori, T. Kawabata, and K. Shikano, “Phoneme segmentation using spectrogram reading knowledge,” in *International Conference on Acoustics, Speech, and Signal Processing, ICASSP*, vol. 1, may 1989, pp. 393 –396.

[13] H. Finster, “Automatic speech segmentation using neural network and phonetic transcription,” in *International Joint Conference on Neural Networks, IJCNN*, vol. 4, jun 1992, pp. 734 –736.

[14] F. Malfrere, O. Deroo, and T. Dutoit, “Phonetic alignement : Speech synthesis based vs. hybrid HMM/ANN,” in *Proceedings of International Conference on Spoken Language Processing, ICSLP*, Sidney, Australia, 1998, pp. 1571–1574.

[15] D. Toledano, “Neural network boundary refining for automatic speech segmentation,” in *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing, ICASSP*, vol. 6, 2000, pp. 3438 –3441.

[16] K.-S. Lee, “MLP-based phone boundary refining for a TTS database,” *IEEE Transactions on Audio, Speech amp; Language Processing*, vol. 14, no. 3, pp. 981–989, 2006.

[17] Y. Suh and Y. Lee, “Phoneme segmentation of continuous speech using multi-layer perceptron,” in *Proceedings of Fourth International Conference on Spoken Language, ICSLP 96*, vol. 3, oct 1996, pp. 1297 –1300.

[18] A. K. Halberstadt, “Heterogeneous acoustic measurements and multiple classifiers for speech recognition,” Ph.D. dissertation, Massachusetts Institute of Technology, MIT, 1998.

[19] L. F. Lamel, R. H. Kassel, and S. Seneff, “Speech database development: Design and analysis of the acoustic-phonetic corpus,” in *Speech Input/Output Assessment and Speech Databases, SIOA*, vol. 2, 1989, pp. 1297 –1300.