Subphonetic Modeling for Speech Recognition

Mei-Yuh Hwang *Xuedong Huang*

School of Computer Science
Carnegie Mellon University
Pittsburgh, Pennsylvania 15213

ABSTRACT

How to capture important acoustic clues and estimate essential parameters reliably is one of the central issues in speech recognition, since we will never have sufficient training data to model various acoustic-phonetic phenomena. Successful examples include subword models with many smoothing techniques. In comparison with subword models, subphonetic modeling may provide a finer level of details. We propose to model subphonetic events with Markov states and treat the state in phonetic hidden Markov models as our basic subphonetic unit — *senone*. A word model is a concatenation of state-dependent senones and senones can be shared across different word models. Senones not only allow parameter sharing, but also enable pronunciation optimization and new word learning, where the phonetic baseform is replaced by the senonic baseform. In this paper, we report preliminary subphonetic modeling results, which not only significantly reduced the word error rate for speaker-independent continuous speech recognition but also demonstrated a novel application for new word learning.

1 INTRODUCTION

For large-vocabulary speech recognition, we will never have sufficient training data to model all the various acoustic-phonetic phenomena. How to capture important acoustic clues and estimate essential parameters reliably is one of the central issues in speech recognition. To share parameters among different word modes, context-dependent subword models have been used successfully in many state-of-the-art speech recognition systems [1, 2, 3, 4]. The principle of parameter sharing can also be extended to subphonetic models. For subphonetic modeling, fenones [5, 6] have been used as the front end output of the IBM acoustic processor. To generate a fenonic pronunciation, multiple examples of each word are obtained. The fenonic baseform is built by searching for a sequence of fenones which has the maximum probability of generating all the given multiple utterances. The codeword-dependent fenonic models are then trained just like phonetic models. We believe that the 200 codeword-dependent fenones may be insufficient for large-vocabulary continuous speech recognition.

In this paper, we propose to model subphonetic events with Markov states. We will treat the state in hidden Markov models (HMMs) as a basic subphonetic unit — *senone*. The total number of HMM states in a system is often too large to be well trained. To reduce the number of free parameters, we can cluster the state-dependent output distributions. Each clustered output distribution is denoted as a senone. In this way, senones can be shared across different models as illustrated in Figure 1.

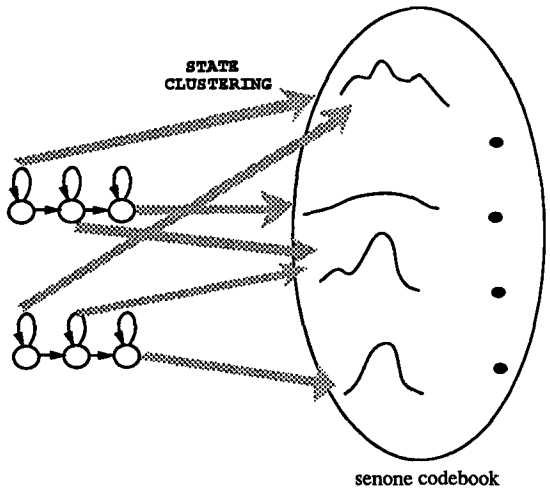

Figure 1: Creation of the senone codebook.

The advantages of senones include better parameter sharing and improved pronunciation optimization. After clustering, different states in different models may share the same senone if they exhibit acoustic similarity. Clustering at the granularity of the state rather than the entire model (like generalized triphones) can keep the dissimilar states of two similar models apart while the other corresponding states are merged, and thus lead to better parameter sharing. For instance, the first, or the second states of the */ey/* phones in PLaCE and RELaTION may be tied together. However, to magnify the acoustic effects of the right contexts, their last states may be kept separately. In addition to finer parameter sharing, senones also give us the freedom to use a larger number of states for each phonetic model. Although an increase in the number of states will increase the total number of free parameters, with senone sharing we can essentially eliminate those redundant states and have the luxury of maintaining the necessary ones.

Since senones depend on Markov states, the senonic baseform of a word can be constructed naturally with the forward-backward algorithm [7]. Regarding pronunciation optimization as well as new word learning, we can use the forward-backward algorithm to iteratively optimize a senone sequence appropriate for modeling multiple utterances of a word. That is, given the multiple examples, we can train a word HMM with the forward-backward algorithm. When the reestimation

reaches its optimality, the estimated states can be *quantized* with the codebook of senones. The closest one can be used to label the state of the word HMM. This sequence of senones becomes the senonic baseform of the word. Here arbitrary sequences of senones are allowed to provide the freedom for the automatically learned pronunciation. After the senonic baseform of every word is determined, the senonic word models may be trained, resulting in a new set of senones. Although each senonic word model generally has more states than the traditional phoneme-concatenated word model, the number of parameters remains the same since the size of the senone codebook is intact.

In dictation applications, new words will often appear during user's usage. A natural extension for pronunciation optimization is to generate senonic baseforms for new words. Automatic determination of phonetic baseforms has been considered by [8], where four utterances and spelling-to-sound rules are need. For the senonic baseform, we can derive senonic baseform only using acoustic data without any spelling information. This is useful for acronym words like IEEE (pronounced as *I-triple-E*), CAT-2 (pronounced as *cat-two*) and foreign person names, where spelling-to-sound rules are hard to generalize. The acoustic-driven senonic baseform can also capture pronunciation of each individual speaker, since in dictation applications, multiple new-word samples are often from the same speaker.

By constructing senone codebook and using senones in the triphone system, we were able to reduce the word error rate of the speaker-independent Resource Management task by 20% in comparison with the generalized triphone [2]. When senones were used for pronunciation optimization, our preliminary results gave us another 15% error reduction in a speaker-independent continuous spelling task. The word error rate was reduced from 11.3% to 9.6%. For new word learning, we used 4 utterances for each new word. Our preliminary results indicate that the error rate of automatically generated senonic baseform is comparable to that of handwritten phonetic baseform.

2 SHARED DISTRIBUTION MODELS

In phone-based HMM systems, each phonetic model is formed by a sequence of states. Phonetic models are shared across different word models. In fact, the state can also be shared across different phonetic models. This section will describe the usage of senones for parameter sharing.

2.1 Senone Construction by State Clustering

The number of triphones in a large vocabulary system is generally very large. With limited training data, there is no hope to obtain well-trained models. Therefore, different technologies have been studied to reduce the number of parameters [1, 9, 2, 10, 11]. In generalized triphones, every state of a triphone is merged with the corresponding state of another triphone in the same cluster. It may be true that some states are merged not because they are similar, but because the other states of the involved models resemble each other. To fulfill more accurate modeling, states with differently-shaped output distributions should be kept apart, even though the other states of the models are tied. Therefore, clustering should be carried out at the output-distribution level rather than the model level. The distribution clustering thus creates a senone codebook as Figure 2 shows [12]. The clustered distributions or senones are fed back to instantiate phonetic models. Thus, states of different phonetic models may share the same senone. This is the same as the *shared-distribution model* (SDM) [13]. Moreover, different states within the same model may also be tied together if too many states are used to model this phone's acoustic variations or if a certain acoustic event appears repetitively within the phone.

1. All HMMs are first estimated.
2. Initially, every output distribution of all HMMs is created as a cluster.
3. Find the most similar pair of clusters and merge them together.
4. For each element in each cluster of the current configuration, move it to another cluster if that results in improvement. Repeat this shifting until no improvement can be made.
5. Go to step 3 unless some convergence criterion is met.

Figure 2: The construction of senones (clustered output distributions).

Senones also give us the freedom to use a larger number of states for each phonetic model. Although an increase in the number of states will increase the total number of free parameters, yet by clustering similar states we can essentially eliminate those redundant states and have the luxury to maintain the necessary ones [13].

2.2 Performance Evaluation

We incorporated the above distribution clustering technique in the SPHINX-II system [14] and experimented on the speaker-independent DARPA Resource Management (RM) task with a word-pair grammar of perplexity 60. The test set consisted of the February 89 and October 89 test sets, totaling 600 sentences. Table 1 shows the word error rates of several systems.

System	Word Error	% Error Reduction
Generalized Triphone	4.7%	–
3500-SDM	4.2%	11%
4500-SDM	3.8%	20%
5500-SDM	4.1%	13%

Table 1: Results of the generalized triphone vs. the SDM.

In the SPHINX system, there were 1100 generalized triphones, each with 3 distinct output distributions. In the

SPHINX-II system, we used 5-state Bakis triphone models and clustered all the output distributions in the 7500 or so triphones down to 3500–5500 senones. The system with 4500 senones had the best performance with the given 3990 training sentences. The similarity between two distributions was measured by their entropies. After two distributions are merged, the entropy-increase, weighted by counts, is computed:

$$(C_a + C_b)H_{a+b} - C_aH_a - C_bH_b$$

where C_a is the summation of the entries of distribution a in terms of counts, and H_a is the entropy. The less the entropy-increase is, the closer the two distributions are. Weighting entropies by counts enables those distributions with less occurring frequency be merged before frequent ones. This makes each senone (shared distribution) more trainable.

2.3 Behavior of State Clustering

To understand the quality of the senone codebook, we examined several examples in comparison with 1100 generalized triphone models. As shown in Figure 3, the two */ey/* triphones in –PLaCE and –LaTION were mapped to the same generalized triphone. Similarly, phone */d/* in StART and AStORIA were mapped to another generalized triphone. Both has the same left context, but different right contexts. States with the same color were tied to the same senone in the 4500-SDM system. x, y, z, and w represent different senones. Figure (a) demonstrates that distribution clustering can keep dissimilar states apart while merging similar states of two models. Figure (b) shows that redundant states inside a model can be squeezed. It also reveals that distribution clustering is able to learn the same effect of similar contexts (*/aa/* and */ao/*) on the current phone (*/d/*).

It is also interesting to note that when 3 states, 5 states, and 7 states per triphone model are used with a senone codebook size of 4500, the average number of distinct senones a triphone used is 2.929, 4.655, and 5.574 respectively. This might imply that 5 states per phonetic model are optimal to model the acoustic variations within a triphone unit for the given DARPA RM training database. In fact, 5-state models indeed gave us the best performance.

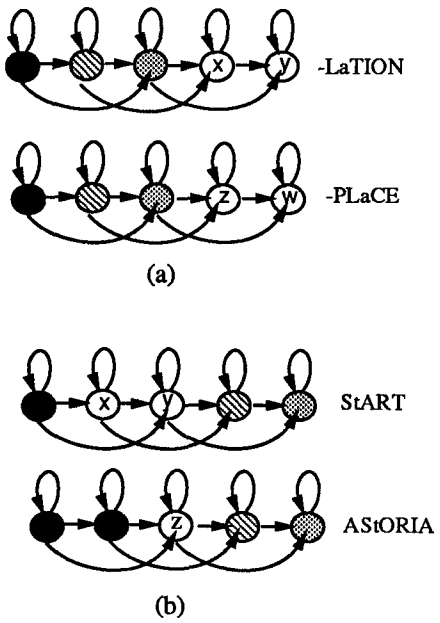

Figure 3: Examples of triphone-clustering and distribution-clustering. Figure (a) shows two */ey/* triphones which were in the same generalized triphone cluster; (b) shows two */d/* triphones in another cluster. In each sub-figure, states with the same color were tied to the same senone. x, y, z, and w represent different senones.

3 PRONUNCIATION OPTIMIZATION

As shown in Figure 1, senones can be shared not only by different phonetic models, but also by different word models. This section will describe one of the most important applications of senones: word pronunciation optimization.

3.1 Senonic Baseform by State Quantization

Phonetic pronunciation optimization has been considered by [15, 8]. Subphonetic modeling also has a potential application to pronunciation learning. Most speech recognition systems use a fixed phonetic transcription for each word in the vocabulary. If a word is transcribed improperly, it will be difficult for the system to recognize it. There may be quite a few improper transcriptions in a large vocabulary system for the given task. Most importantly, some words may be pronounced in several different ways such as THE (/dh *ax*/ or /dh *ih*/), TOMATO (/t ax m *ey* dx ow/ or /t ax m *aa* dx ow/), and so on. We can use multiple phonetic transcriptions for every word, or to learn the pronunciation automatically from the data.

Figure 4 shows the algorithm which looks for the most appropriate senonic baseform for a given word when training examples are available.

1. Compute the average duration (number of time-frames), given multiple tokens of the word.
2. Build a Bakis word HMM with the number of states equal to a portion of the average duration (usually 0.8).
3. Run several iterations (usually 2 – 3) of the forward-backward algorithm on the word model starting from uniform output distributions, using the given utterance tokens.
4. Quantize each state of the estimated word model with the senone codebook.

Figure 4: The determination of a senonic baseform, given multiple training tokens.

Here arbitrary sequences of senones are allowed to provide the freedom for the automatically learned pronunciation. This senonic baseform tightly combines the model and acoustic data. After the senonic baseform of every word is determined,

the senonic word models may be trained, resulting in a new set of senones.

Similar to fenones, senones take full advantage of the multiple utterances in baseform construction. In addition, both phonetic baseform and senonic baseform can be used together, without doubling the number of parameters in contrast to fenones. So we can keep using phonetic baseform when training examples are unavailable. The senone codebook also has a better acoustic resolution in comparison with the 200 VQ-dependent fenones. Although each senonic word model generally has more states than the traditional phoneme-concatenated word model, the number of parameters are not increased since the size of the senone codebook is fixed.

3.2 Performance Evaluation

As a pivotal experiment for pronunciation learning, we used the speaker-independent continuous spelling task (26 English alphabet). No grammar is used. There are 1132 training sentences from 100 speakers and 162 testing sentences from 12 new speakers. The training data were segmented into words by a set of existing HMMs and the Viterbi alignment [16, 1]. For each word, we split its training data into several groups by a DTW clustering procedure according to their acoustic resemblance. Different groups represent different acoustic realizations of the same word. For each word group, we estimated the word model and computed a senonic baseform as Figure 4 describes. The number of states of a word model was equal to 75% of the average duration. The Euclidean distance was used as the distortion measure during state quantization.

We calculated the predicting ability of the senonic word model $M_{w,g}$ obtained from the g-th group of word w as:

$$\sum_{\mathbf{X}_w \notin \text{group } g} \log P(\mathbf{X}_w | M_{w,g}) \; / \sum_{\mathbf{X}_w \notin \text{group } g} |\mathbf{X}_w|$$

where $\mathbf{X}_w$ is an utterance of word w.

For each word, we picked two models that had the best predicting abilities. The pronunciation of each word utterance in the training set was labeled by:

$$model(\mathbf{X}_w) = \underset{M_{w,g} \in top2}{\text{argmax}} \{ P(\mathbf{X}_w | M_{w,g}) \}$$

After the training data were labeled in this way, we retrained the system parameters by using the senonic baseform. Table 2 shows the word error rate. Both systems used the sex-dependent semi-continuous HMMs. The baseline used word-dependent phonetic models. Therefore, it was essentially a word-based system. Fifty-six word-dependent phonetic models were used. Note both systems used exactly the same number of parameters.

This preliminary results indicated that the senonic baseform can capture detailed pronunciation variations for speaker-independent speech recognition.

System	Word Error	% Error Reduction
phonetic baseform	11.3%	–
senonic baseform	9.6%	15%

Table 2: Results of the phonetic baseform vs. the senonic baseform on the spelling task.

4 NEW WORD LEARNING

In dictation applications, we can start from speaker-independent system. However, new words will often appear when users are dictating. In real applications, these new word samples are often speaker-dependent albeit speaker-independent systems may be used initially. A natural extension for pronunciation optimization is to generate speaker-dependent senonic baseforms for these new words. In this study, we assume possible new words are already detected, and we want to derive the senonic baseforms of new words automatically. We are interested in using acoustic data only. This is useful for acronym words like IEEE (pronounced as *I-triple-E*), CAT-2 (pronounced as *cat-two*) and foreign person names, where spelling-to-sound rules are hard to generalize. The senonic baseform can also capture pronunciation characteristics of each individual speaker that cannot be represented in the phonetic baseform.

4.1 Experimental Database and System Configuration

With word-based senonic models, it is hard to incorporate between-word co-articulation modeling. Therefore, our baseline system used within-word triphone models only. Again we chose RM as the experimental task. Speaker-independent (SI) sex-dependent SDMs were used as our baseline system for this study. New word training and testing data are speaker-dependent (SD). We used the four speakers (2 females, 2 males) from the June-1990 test set; each supplied 2520 SD sentences. The SD sentences were segmented into words using the Viterbi alignment.

Then we chose randomly 42 words that occurred frequently in the SD database (so that we have enough testing data) as shown in Table 3, where their frequencies in the speaker-independent training database are also included. For each speaker and each of these words, 4 utterances were used as samples to learn the senonic baseform, and at most 10 other utterances as testing. Therefore, the senonic baseform of a word is speaker-dependent. There were together 1460 testing word utterances for the four speakers. During recognition, the segmented data were tested in an isolated-speech mode without any grammar.

4.2 State Quantization of the Senonic Baseform

For each of the 42 words, we used 4 utterances to construct the senonic baseform. The number of states was set to be 0.8 of the average duration. To quantize states at step 4 of Figure 4, we aligned the sample utterances against the estimated word model by the Viterbi algorithm. Thus, each state had 5 to 7 frames on average. Each state of the word model is quantized to the senone that has the maximum probability of generating all the aligned frames. Given a certain senone, $senone$, the probability of generating the aligned frames of state s is computed in the same manner as the semi-continuous output probability:

word	SI(F/M) training	word	SI(F/M) training
AAW	15/28	JAPAN	10/28
ARCTIC	7/24	LATITUDE	22/49
ASUW	13/27	LATITUDES	12/27
ASW	11/30	LINK-11	6/20
AVERAGE	33/70	LONGITUDE	24/46
C1	12/33	LONGITUDES	13/23
C2	17/33	MAX	15/23
C3	16/41	MAXIMUM	24/55
C4	14/33	MIW	16/31
C5	11/44	MOB	10/29
CAPABLE	34/99	MOZAMBIQUE	13/28
CAPACITY	3/75	NTDS	10/26
CASREP	3/9	NUCLEAR	6/15
CASREPED	2/4	PACFLT	3/8
CASREPS	11/8	PEARL-HARBOR	2/6
CASUALTY	42/88	PROPULSION	15/21
CHINA	12/27	READINESS	59/136
FLEET	39/96	SOLOMON	12/24
FLEETS	2/9	STRAIT	26/77
FORMOSA	9/29	THAILAND	13/26
INDIAN	9/29	TOKIN	13/27

Table 3: New words and their frequencies in the speaker-independent training set (Female/Male).

$$Pr(\mathbf{X}|senone) = \prod_{\forall \mathbf{X}_i \ aligned\ to\ s} Pr(\mathbf{x}_i|senone)$$

$$= \prod_{\forall \mathbf{X}_i \ aligned\ to\ s} \sum_{k=1}^{L} \mathbf{b}(k|senone) f_k(\mathbf{x}_i)$$

where $\mathbf{b}(.|senone)$ denote the discrete output distribution that represents *senone*, L denotes the size of the front-end VQ codebook, and $f_k(.)$ denote the probability density function of codeword k.

4.3 Experimental Performance

For the hand-written phonetic baseform, the word error rate was 2.67% for the 1460 word utterances. As a pilot study, a separate senonic baseform was constructed for CASREP and its derivatives (CASREPED, and CASREPS). Similarly, for the singular and plural forms of the selected nouns. The selected 42 words were modeled by automatically constructed senonic baseforms. They are used together with the rest 955 words (phonetic baseforms) in the RM task. The word error rate was 6.23%. Most of the errors came from the derivative confusion.

To reduce the derivative confusion, we concatenated the original senonic baseform with the possible suffix phonemes as the baseform for the derived words. For example, the baseform of FLEETS became */fleet <ts s ix-z>/*, where the context-independent phone model */ts/*, */s/*, and the concatenated */ix z/* were appended parallelly after the senonic baseform of FLEET. In this way, no training data were used to learn the pronunciations of the derivatives. This *suffix senonic* approach significantly reduced the word error to 3.63%. Still there were a lot of misrecognitions of CASREPED to be CASREP and MAX to be NEXT. These were due to the high confusion between */td/* and */pd/*, */m/* and */n/*. The above results are summarized in Table 4.

system	error rate
hand-written phonetic baseform	2.67%
pilot senonic baseform	6.23%
suffix senonic baseform	3.63%

Table 4: Results of the senonic baseforms on the 1460 word utterances for the selected 42 words.

The study reported here is preliminary. Refinement on the algorithm of senonic-baseform construction (especially incorporation of the spelling information) is still under investigation. Our goal is to approach the phonetic system.

5 CONCLUSION

In this paper, we developed the framework of senones — state-dependent subphonetic unit. Senones are created by *clustering* states of triphone models. Thus, we reduced the the number of system parameters with the senone codebook, which renders finer acoustic modeling and provides a way to learn the model topology. In the mean time, we can construct senonic baseforms to improve phonetic baseforms and learn new words without enlarging system parameters. Senonic baseforms are constructed by *quantizing* the states of estimated word models with the senone codebook. We demonstrated that senones can not only significantly improve speaker-independent continuous speech recognition but also have a novel application for new word learning.

Acknowledgements

This research was sponsored by the Defense Advanced Research Projects Agency (DOD), Arpa Order No. 5167, under contract number N00039-85-C-0163. The authors would like to express their gratitude to Professor Raj Reddy for his encouragement and support, and other members of CMU speech group for their help.

References

[1] Schwartz, R., Chow, Y., Kimball, O., Roucos, S., Krasner, M., and Makhoul, J. *Context-Dependent Modeling for Acoustic-Phonetic Recognition of Continuous Speech.* in: **IEEE International Conference on Acoustics, Speech, and Signal Processing.** 1985, pp. 1205–1208.

[2] Lee, K. *Context-Dependent Phonetic Hidden Markov Models for Continuous Speech Recognition.* **IEEE Transactions on Acoustics, Speech, and Signal Processing,** April 1990, pp. 599–609.

[3] Lee, C., Giachin, E., Rabiner, R., L. P., and Rosenberg, A. *Improved Acoustic Modeling for Continuous Speech Recognition.* in: **DARPA Speech and Language Workshop**. Morgan Kaufmann Publishers, San Mateo, CA, 1990.

[4] Bahl, L., de Souza, P., Gopalakrishnan, P., Nahamoo, D., and Picheny, M. *Decision Trees for Phonological Rules in Continuous Speech.* in: **IEEE International Conference on Acoustics, Speech, and Signal Processing**. 1991, pp. 185–188.

[5] Bahl, L., Brown, P., de Souza, P., and Mercer, R. a. *A Method for the Construction of Acoustic Markov Models for Words.* no. RC 13099 (#58580), IBM Thomas J. Watson Research Center, September 1987.

[6] Bahl, L., Brown, P., De Souza, P., and Mercer, R. *Acoustic Markov Models Used in the Tangora Speech Recognition System.* in: **IEEE International Conference on Acoustics, Speech, and Signal Processing**. 1988.

[7] Bahl, L. R., Jelinek, F., and Mercer, R. *A Maximum Likelihood Approach to Continuous Speech Recognition.* **IEEE Transactions on Pattern Analysis and Machine Intelligence**, vol. PAMI-5 (1983), pp. 179–190.

[8] Bahl, L. and et. al. *Automatic Phonetic Baseform Determination.* in: **IEEE International Conference on Acoustics, Speech, and Signal Processing**. 1991, pp. 173–176.

[9] Schwartz, R., Kimball, O., Kubala, F., Feng, M., Chow, Y., C., B., and J., M. *Robust Smoothing Methods for Discrete Hidden Markov Models.* in: **IEEE International Conference on Acoustics, Speech, and Signal Processing**. 1989, pp. 548–551.

[10] Paul, D. *The Lincoln Tied-Mixture HMM Continuous Speech Recognizer.* in: **DARPA Speech and Language Workshop**. Morgan Kaufmann Publishers, San Mateo, CA, 1990, pp. 332–336.

[11] Huang, X., Ariki, Y., and Jack, M. **Hidden Markov Models for Speech Recognition**. Edinburgh University Press, Edinburgh, U.K., 1990.

[12] Lee, K., Hon, H., and Reddy, R. *An Overview of the SPHINX Speech Recognition System.* **IEEE Transactions on Acoustics, Speech, and Signal Processing**, January 1990, pp. 35–45.

[13] Hwang, M. and Huang, X. *Acoustic Classification of Phonetic Hidden Markov Models.* in: **Proceedings of Eurospeech**. 1991.

[14] Huang, X., Alleva, F., Hon, H., Hwang, M., and Rosenfeld, R. *The SPHINX-II Speech Recognition System: An Overview.* Technical Report, no. CMU-CS-92-112, School of Computer Science, Carnegie Mellon University, Pittsburgh, PA, February 1992.

[15] Bernstein, J., Cohen, M., Murveit, H., and Weintraub, M. *Linguistic Constraints in Hidden Markov Model Based Speech Recognition.* in: **IEEE International Conference on Acoustics, Speech, and Signal Processing**. 1989.

[16] Viterbi, A. J. *Error Bounds for Convolutional Codes and an Asymptotically Optimum Decoding Algorithm.* **IEEE Transactions on Information Theory**, vol. IT-13 (1967), pp. 260–269.