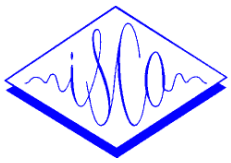

3rd International Conference on
Spoken Language Processing (ICSLP 94)
Yokohama, Japan
September 18-22, 1994

# SPEECH RECOGNITION WITHOUT GRAMMAR OR VOCABULARY CONSTRAINTS

*Harald Singer and Jun-ichi Takami*

ATR Interpreting Telecommunications Res. Labs.
2-2 Hikaridai, Seika-cho, Soraku-gun, Kyoto 619-02 Japan
E-mail: singer@itl.atr.co.jp

## ABSTRACT

Out-of-vocabulary words and ungrammatical utterances are two major problems in speech recognition. We believe that improving the acoustic model is essential in dealing with these problems. We propose to use a 'phonetic typewriter' as an evaluation method. Unlike common approaches, which evaluate acoustic and language model together, this allows direct evaluation of the acoustic model. A comparison of context-independent phone models based on continuous mixture HMM (20 mixtures per state) with context-dependent phone models based on HMnet[4] (3 mixtures per state) showed that phoneme error rate can be halved by using the latter models. The same 'phonetic typewriter' paradigm can also be used directly as a speech recognition method, in which speech is recognized as a string of phonemes without constraints on vocabulary or grammar. We show that over 97 % phoneme recognition accuracy can be achieved if our best acoustic model is used.

## 1. INTRODUCTION

A major advantage of speech as an efficient man-machine interface compared to pointing device based interfaces (e.g. mouse) lies in its potentially unlimited nature. Humans use speech because it allows them freedom of expression. Using this freedom, however, entails the typical phenomena of non-read speech such as ungrammatical utterances, false starts, hesitations and filled-pauses. Our long-term goal is the development of a speech recognition system that deals gracefully with these phenomena.

The development of speech recognition systems which can deal with these phenomena necessitates better evaluation methods for the underlying acoustic models. Previously, acoustic models were mainly evaluated using phoneme-labeled databases with explicit phoneme boundaries or through full-fledged speech recognition systems using task-dependent grammars. Both of these methods are clearly insufficient as in the former method only the discriminative power, given phoneme segmentation boundaries, is compared and in the latter method the constraints of the language model have a major influence on the result. The former method also needs a phoneme-labeled database, whose creation is expensive and time-consuming.

As we want to measure both the discriminative power and the segmentation abilities of the acoustic models in question, we propose a 'phonetic typewriter' as a task-independent evaluation method for acoustic models. Evaluation of phoneme models without labels has also been proposed recently[3], but that algorithm does not take deletions and insertions into account.

Because of the advances in acoustic modeling, we are also investigating the possibility of using the 'phonetic typewriter' as a grammar-, task-, and vocabulary-independent recognition front end.

In section 2 we will first discuss the major constituents of this research, i.e. the acoustic model, the language model and the search strategy. Detailed experimental results are given in section 3 and finally conclusions are drawn in section 4.

## 2. CONSTITUENT TECHNOLOGIES

### 2.1. Acoustic Model

Recently, one of the authors proposed the HMnet, a context-dependent acoustic model[4]. A sketch is shown in Fig. 1.

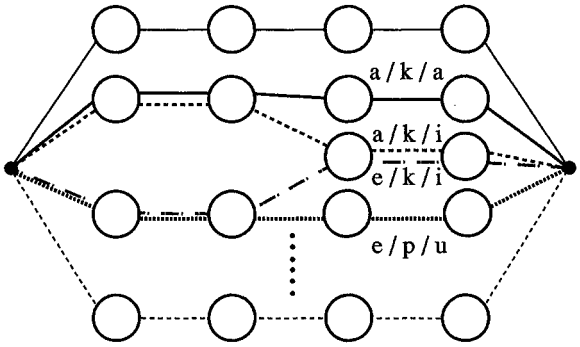


Figure 1: sketch of an HMnet

The same phoneme can have different acoustic realizations according to its context, e.g. preceding phoneme and succeeding phoneme. To cover these allophonic variations context-dependent phone models have been proposed and shown to be superior to context-independent phoneme models. The HMnet is one type of context-dependent phoneme model which has the following features:

- Topology of the HMnet is automatically trained by the Successive State Splitting algorithm (SSS) [4], which thus does not require ad-hoc decisions about similar phonetic contexts.
- Due to state sharing between different allophones, a large number of allophones can be represented by comparatively few states and thus a high degree of robustness can be achieved.

The HMnet's we use here can represent 1000 $\sim$ 2000 allophones with only 400 $\sim$ 600 states. Representing such a number of allophones without state sharing would require 4000 $\sim$ 7000 states. The HMnet is thus a concise representation. The (always) limited amount of training data is used efficiently and thus allows robust estimation of the model parameters, which is also very important for speaker adaptation [5].

## 2.2. Language Constraints

Table 1: Phonotactic constraints in Japanese

| phoneme | possible succeeding phoneme |
|---|---|
| b | y, a, i, u, e, o |
| d | a, i, e, o |
| g | y, a, i, u, e, o |
| p | y, a, i, u, e, o |
| t | a, i, e, o |
| k | y, a, i, u, e, o |
| m | y, a, i, u, e, o |
| n | y, a, i, u, e, o |
| N | b, d, g, p, t, k, m, n, s, sh, h, z, ch, ts, zh, r, w, y, a, i, u, e, o, - |
| s | a, u, e, o |
| sh | y, i |
| h | y, a, i, u, e, o |
| z | a, u, e, o |
| ch | y, i |
| ts | u |
| zh | y, i |
| r | y, a, i, u, e, o |
| w | a |
| y | a, u, o |
| a | b, d, g, p, t, k, m, n, N, s, sh, h, z, ch, ts, zh, r, w, y, a, i, u, e, o, q, – |
| i | b, d, g, p, t, k, m, n, N, s, sh, h, z, ch, ts, zh, r, w, y, a, i, u, e, o, q, – |
| u | b, d, g, p, t, k, m, n, N, s, sh, h, z, ch, ts, zh, r, w, y, a, i, u, e, o, q, – |
| e | b, d, g, p, t, k, m, n, N, s, sh, h, z, ch, ts, zh, r, w, y, a, i, u, e, o, q, – |
| o | b, d, g, p, t, k, m, n, N, s, sh, h, z, ch, ts, zh, r, w, y, a, i, u, e, o, q, – |
| q | p, t, k, s, sh, ch, ts |
| – | b, d, g, p, t, k, m, n, s, sh, h, z, ch, ts, zh, r, w, y, a, i, u, e, o, – |

As stated in the introduction, the evaluation of the acoustic models should be decoupled from the qualities of the language model. The reason for this is that eventually the acoustic models will be used with other language models and even other tasks. If we knew that our acoustic models were only used with a certain task-dependent language model, use of this language model in evaluating the acoustic models would even be advantageous. However, as we want to develop general training algorithms and not a certain application, this task-dependent language model would make a detailed error analysis impossible.

On the other hand, if no constraints on possible phoneme sequences are applied, the search space of possible recognition candidates becomes too large. We are thus looking for some "general" language model that can be used for every imaginable task.

The language model we choose models the phonotactic constraints of Japanese. Phoneme sequences that cannot appear in Japanese are excluded from the search, i.e. this implements the lowest level of linguistic constraints. There is no penalty on rare occurrences of phoneme sequences, a feature which is desirable for a phonetic typewriter. The constraints are implemented as the simple rules shown in Tbl. 1, which is basically a list of possible phoneme connections in Japanese, or in other words a phoneme pair grammar. The phone perplexity of this "grammar" is about 10.

In [2], the weak constraints of syllable trigrams were used as "general" language model. There are at least two problems with this method: first, the probabilities depend strongly on the data which were used for training the trigrams. For example, syllable sequences which do not appear in the training data are heavily penalized. Secondly, even if these probabilities were the true probabilities, rare occurrences would still be penalized. For error minimization in a speech recognizer, this would have no particularly detrimental effect. However, this would somehow distort the evaluation of acoustic models with regard to rare phoneme sequences.

## 2.3. Search Strategy

Search is performed with a frame-synchronous One-Pass algorithm using $N$-best candidates. Merging of hypotheses, which differ only in phoneme boundary position or in the number of identical repeated phonemes is also implemented.

### *2.3.1. N-best Candidates*

An $N$-best One-Pass algorithm can be easily implemented by storing at each grid point not only the best accumulated score but the top $N$-best candidates. In a straightforward implementation many candidates will differ only in the phoneme boundaries. The purpose of the $N$-best search, however, is to keep as many hypotheses as possible that differ in phoneme sequence.

Therefore, at every grid point, only the $N$-best partial results with different phoneme sequences are chosen.

### *2.3.2. Merging of Candidates with Phoneme Repetitions*

Another problem with the $N$-best search is the appearance of multiple candidates which differ only in the number of repetitions of the same stationary phonemes.

As shown in Tbl. 1, an infinite number of repetitions are allowed for the phonemes /a/, /i/, /u/, /e/, /o/ and the silence model /–/. Therefore, many candidates are generated which differ only in the number of repeated stationary sounds like 「もし /m o sh i/」 and 「申し /m o o sh i/」. It is doubtful that the number of repetitions of stationary sounds can be decided by acoustic features only, and it would be highly inefficient to keep these multiple candidates.

At each gridpoint, candidates who differ only in the number of repetitions of stationary sounds are therefore merged. Information about the repetitions of stationary sounds is thus lost but we can cope with this by post processing as the phoneme boundaries are also stored.

### *2.3.3. Search Algorithm*

The standard One-Pass algorithm keeps a back pointer at each grid point so that at the end of the forward calculation a traceback puts out the optimal phoneme sequence.

In our implementation, comparisons between different hypotheses have to be performed frequently and it would be too complex to do a traceback every time a comparison becomes necessary. We therefore put the current phoneme sequence including the phoneme boundary information in a list structure for every grid point and can thus avoid traceback calculation at the expense of some increased memory usage[1].

The detailed algorithm is as follows:

[definitions]

$T$ : number of input frames
$J$ : expanded state number
$N$ : depth of $N$-best search
$M$ : beamwidth
$\mathbf{I}_j$ : set of possible succeeding states for state $j$
$\pi_0$ : set of possible start states
$\pi_f$ : set of possible final states
$a_{ij}$ : transition probability from state $i$ to state $j$

$x_t$ : observation at frame $t$
$b_i(x_t)$ : output probability at state $i$ for observation $x_t$
$q_i$ : phoneme label at state $i$
$S(t,j,n)$ : accumulated score for frame $t$, state $j$, rank $n$
$\mathbf{P}(t,j,n) = \{p_{t,j,n,l}\}$ :
list of phoneme labels for frame $t$, state $j$, rank $n$
$\mathbf{D}(t,j,n) = \{d_{t,j,n,l}\}$ :
list of phone durations for frame $t$, state $j$, rank $n$
$L(t,j,n)$ : number of elements of $\mathbf{P}(t,j,n)$ or $\mathbf{D}(t,j,n)$

(1) initialization

$$\begin{cases} S(0,j,1) \leftarrow 0.0 \\ \mathbf{P}(0,j,1) \leftarrow \{q_j\} \\ \mathbf{L}(0,j,1) \leftarrow \{0\} \\ L(0,j,1) \leftarrow 1 \end{cases} \quad ( j \in \pi_0 )$$

$$\begin{cases} S(0,j,n) \leftarrow -\infty \\ \mathbf{P}(0,j,n) \leftarrow \phi \\ \mathbf{L}(0,j,n) \leftarrow \phi \\ L(0,j,n) \leftarrow 0 \end{cases} \quad \left( \begin{array}{c} j \notin \pi_0 \ and \ n = 1 \\ or \\ 2 \le n \le N \end{array} \right)$$

(2) for $t = 1,2,\cdots T$ do (3) $\sim$ (11)
(3) if $t < T$ for $j = 1,2,\cdots J$, if $t = T$ for $j = J+1$, do (4) $\sim$ (10)
(4) if $t < T$ for $i \in \mathbf{I}_j$, if $t = T$ for $i \in \pi_f$, do (5) $\sim$ (7)
(5) for $n = 1,2,\cdots N$ do (6), (7)
(6) if $S(t-1,i,n) = -\infty$,
$S'(t,i,j,n) \leftarrow -\infty$, return to (4)
else,
$S'(t,i,j,n) \leftarrow S(t-1,i,n) + \log b_i(x_t) + \log a_{ij}$
(7) if $t = T$, or $q_i = q_j$,

$$\begin{cases} \mathbf{P}'(t,i,j,n) \leftarrow \mathbf{P}(t-1,i,n) \\ \mathbf{D}'(t,i,j,n) \leftarrow \{d_{t-1,i,n,1}, \cdots d_{t-1,i,n,L(t-1,i,n)-1}, \\ \qquad d_{t-1,i,n,L(t-1,i,n)} + 1\} \\ L'(t,i,j,n) \leftarrow L(t-1,i,n) \end{cases}$$

if $q_i \neq q_j$,

$$\begin{cases} \mathbf{P}'(t,i,j,n) \leftarrow \{p_{t-1,i,n,1}, \cdots p_{t-1,i,n,L(t-1,i,n)}, q_j\} \\ \mathbf{D}'(t,i,j,n) \leftarrow \{d_{t-1,i,n,1}, \cdots d_{t-1,i,n,L(t-1,i,n)-1}, \\ \qquad d_{t-1,i,n,L(t-1,i,n)} + 1, 0\} \\ L'(t,i,j,n) \leftarrow L(t-1,i,n) + 1 \end{cases}$$

(8) for $m = 1,2,\cdots N$ do (9), (10)
(9) $(\hat{i},\hat{n}) = \underset{i,\ n}{\operatorname{argmax}} S'(t,i,j,n)$
however, if $m > 1$, exclude $i,n$ so that $\mathbf{P}'(t,i,j,n) = \mathbf{P}(t,j,k)$ $(k = 1,2,\cdots m-1)$

$$\begin{cases} S(t,j,m) \leftarrow S'(t,\hat{i},j,\hat{n}) \\ \mathbf{P}(t,j,m) \leftarrow \mathbf{P}'(t,\hat{i},j,\hat{n}) \\ \mathbf{D}(t,j,m) \leftarrow \mathbf{D}'(t,\hat{i},j,\hat{n}) \\ L(t,j,m) \leftarrow L'(t,\hat{i},j,\hat{n}) \end{cases}$$

(10) $S'(t,\hat{i},j,\hat{n}) \leftarrow -\infty$
(11) beam pruning
for $j = 1,2,\cdots J$, $n = 1,2,\cdots N$, select $M$ largest values from $S(t,j,n)$, set all other values to $-\infty$
(12) recognition results
result for rank $n$ is the phoneme sequence $P(T,J+1,n)$ of length $L(T,J+1,n)$.
duration (in frames) for each phoneme $p_{T,J+1,n,l}$ is $d_{T,J+1,n,l}$.

## 3. EXPERIMENTS

### 3.1. Acoustic Models

To find out which HMnet is the best compromise between number of free parameters (states, distributions per state, mixtures per distribution) and a given amount of training data, 4 different HMnet's are tested. For comparison, we also tested the performance of 3 context-independent models:

(a) HMnet (400 states, single Gaussian)
(b) HMnet (400 states, 3 Gaussian mixtures)
(c) HMnet (600 states, single Gaussian)
(d) HMnet (600 states, 3 Gaussian mixtures)
(e) phoneme HMM (3 states/phoneme, 5 Gaussian mixtures)
(f) phoneme HMM (3 states/phoneme, 10 Gaussian mixtures)
(g) phoneme HMM (3 states/phoneme, 20 Gaussian mixtures)

All distributions are Gaussian with diagonal covariance matrices. Details for each model are given in Tbl. 2. All acoustic models are furthermore adapted to phrase speaking style [5].

Table 2: specification of acoustic models

| acoustic model | total number of distributions | number of allophones | number of expanded states |
|---|---|---|---|
| (a) | 400 | 1034 | 4111 |
| (b) | 1200 | 1034 | 4111 |
| (c) | 600 | 1816 | 7224 |
| (d) | 1800 | 1816 | 7224 |
| (e) | 390 | 26 | 78 |
| (f) | 780 | 26 | 78 |
| (g) | 1560 | 26 | 78 |

### 3.2. Experimental Conditions

The experimental conditions are as follows:

- 1 Japanese male speaker(MHT)

[acoustic preprocessing]

- sampling rate: 12kHz
- sampling precision: 16bit
- frame rate: 5ms
- frame length: 20ms
- window type: Hamming
- acoustic parameters: 34 dimensional vector consisting of 16 order LPC Cepstrum, log power, 16 order $\Delta$LPC Cepstrum, $\Delta$log power

[phoneme labels]

- 26 labels: b, d, g, p, t, k, m, n, N (syllabic nasal), s, sh, h, z, ch, ts, zh, r, w, y, a, i, u, e, o, q (geminate), – (silence)

[perplexity due to phonotactic constraints]

- 10.2

[training data for acoustic model]

- even numbered words of 5240 Japanese words, uttered word by word

[evaluation data]

- international conference registration conversation, 279 phrases uttered phrase by phrase (SB3)
- total number of phonemes: 2684
- average number of phonemes per phrase: 9.62

[data for speaking style adaptation]

- 1098 phrases uttered with similar speaking rate as the evaluation data (SA1, 2, 4 and SB1, 2, 4)

### 3.3. Evaluation for Each Acoustic Model

Tbl. 3 shows the evaluation results for acoustic models (a) $\sim$ (g) with an $N$-best depth of 10 and beam width $\infty$ (full search).

In accord with the reasoning used for merging candidates with repetitions of stationary sounds, we define an *adjusted recognition rate*. Two types of errors are not counted, where we believe there is no meaningful distinction between candidate phoneme strings based on acoustic features alone:

Table 3: experimental evaluation results for each acoustic model type (full search, rates in % )

| acoustic model | rank 1 only | | up to rank 10 | | rank 1 only (total number of phonemes: 2684) | | | | | required CPU time ratio |
|---|---|---|---|---|---|---|---|---|---|---|
| | phrase rate | adjusted phrase rate | phrase rate | adjusted phrase rate | ins. errors | del. errors | subst. errors | phoneme rate | adjusted phoneme rate | |
| (a) | 59.86 | 63.08 | 87.81 | 89.25 | 61 | 24 | 96 | 93.26 | 93.80 | 1.00 |
| (b) | 73.48 | 77.42 | 93.19 | 93.55 | 46 | 12 | 43 | 96.24 | 96.72 | 1.07 |
| (c) | 67.38 | 69.89 | 89.61 | 90.68 | 52 | 14 | 74 | 94.78 | 95.12 | 1.76 |
| (d) | 79.21 | 81.72 | 95.70 | 95.70 | 29 | 10 | 40 | 97.06 | 97.39 | 1.79 |
| (e) | 56.27 | 63.08 | 82.08 | 83.87 | 81 | 31 | 95 | 92.29 | 93.29 | 0.025 |
| (f) | 64.52 | 69.89 | 85.66 | 87.10 | 62 | 30 | 69 | 94.00 | 94.42 | 0.031 |
| (g) | 69.53 | 74.55 | 88.89 | 90.32 | 50 | 19 | 61 | 95.16 | 95.67 | 0.044 |

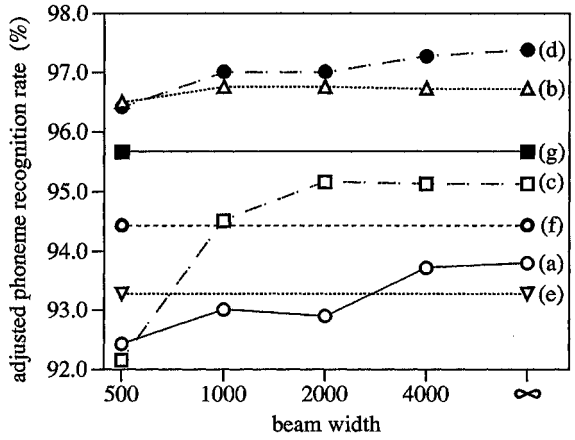


Figure 2: relation between beam width and adjusted phoneme recognition rate

(1) errors due to insertion or deletion of silence /−/ or geminate /q/ before unvoiced plosives /p/, /t/, /k/, /ts/, /ch/ /−/(or geminate /q/) (e.g 「赤 /a k a/」 and 「悪化 /a q k a/」 )

(2) errors due to insertion or deletion of a syllabic nasal /N/ before nasals /m/, /n/ or the voiced velar nasal consonant /g/ (e.g. 「穴 /a n a/」 and 「あんな /a N n a/」 )

Required CPU time ration in Tbl. 3 has been normalized with respect to acoustic model (a).

Comparing results for models with similar number of distributions (e.g. (a) vs. (e), (c) vs. (f), (b) or (d) vs. (g)), the HMnet clearly outperforms the context-independent HMM's. Particularly, (d) is vastly superior to all other models.

On the other hand, the major disadvantage of using context-dependent acoustic models compared to context-independent HMM's is the increase in CPU time for recognition. Especially for the One-Pass algorithm, the CPU time is roughly proportional to the number of expanded states for all models.

### 3.4. Influence of Beam Search

To reduce CPU time while keeping recognition accuracy high, we introduced a beam search technique, i.e. after calculating each frame only the $M$ best values are kept. Relation between beam width $M$ and adjusted phoneme recognition rate for all models is plotted in Fig. 2. For model (d) and a beam width $M$ of 4000 we get a speed-up by a factor of 3 with only a negligible loss in recognition accuracy ($M = 500$ gives a speedup of factor 5 with a 1% loss in accuracy).

## 4. CONCLUSIONS

Good acoustic models are essential for continuous speech recognition and even more so if the task deals with non-read speech as it becomes difficult to apply strong linguistic constraints. To develop better acoustic models we proposed a 'phonetic typewriter' as a task- and language-independent evaluation method. The search space was only constrained by Japanese phonotactics, i.e. a phoneme pair grammar defining which phonemes can follow each other.

For recognition, an $N$-best Viterbi search was used. To constrain computational complexity, we implemented several improvements like merging of boundaries, merging of identical phonemes, efficient book-keeping for the back trace information and efficient sorting strategies at each grid point.

Comparison of context-independent Gaussian mixture phone models with more acoustically precise context-dependent phoneme models (HMnet), generated by the Successive State Splitting algorithm (SSS) [4] showed that the phoneme error rate can be halved using the latter models.

Furthermore, a phoneme recognition accuracy of over 97 % showed that the 'phonetic typewriter' is not just an evaluation method but a realistic alternative to grammar driven systems. Some advantages of this phonetic typewriter paradigm are flexibility in regard to new tasks and new languages (only phonotactic constraints have to be specified and neither have grammars to be written, nor duration models trained). We do not underestimate the difficulty of interfacing with a language understanding system, but in comparison with previous 'phonetic typewriter' attempts, our approach provides a high phoneme recognition accuracy to start with.

## REFERENCES


[1] T. Araki, J. Murakami, and S. Ikehara. Algorithms for deciding a maximum likelihood candidate from syllable lattice using m-th order Markov model. Technical Report CS90-55, IEICE, 1990. (in Japanese).

[2] T. Kawabata, T. Hanazawa, K. Itoh, and K. Shikano. Japanese phonetic typewriter using HMM phone recognition and stochastic phone-sequence modeling. *IEICE TRANSACTIONS*, E74(7):1783–1787, July 1991.

[3] Y. Minami, T. Matsuoka, and K. Shikano. Phoneme HMM evaluation algorithm without phoneme labeling applied to continuous speech HMM evaluation. *IEICE TRANSACTIONS*, J77-A(2):267–273, February 1994. (in Japanese).

[4] J. Takami and S. Sagayama. A successive state splitting algorithm for efficient allophone modeling. In *Proc. ICASSP*, volume 1, pages 573–576, San Francisco, March 1992.

[5] J. Takami and S. Sagayama. A speaker adaptation technique for Hidden Markov networks. Technical Report SP93-50, IEICE, 1993. (in Japanese).