Dynamic approaches for real-time LVCSR

リアルタイム大語彙連続音声認識における動的アプローチ

February 2008

Shoei SATO
Dynamic approaches for real-time LVCSR

February 2008

Department of Computer Science,
Graduate School of Science and Engineering,
WASEDA University

Shoei SATO
# Contents

1 Introduction 1

1.1 Real-time speech recognition for closed captioning 4

1.1.1 Direct method 6

1.1.2 Re-speak method 10

1.2 Speech recognition for segment metadata 13

1.3 Organization of this thesis 15

2 Technical introduction 18

2.1 Overview of a statistical speech recognition 18

2.2 Speech analysis 20

2.2.1 Cepstrum 21

2.2.2 MFCC 22

2.2.3 Short-term modulation 22

2.2.4 Long-term modulation 23

2.2.5 Spectral subtraction 24

2.3 Acoustic model 25

2.3.1 Hidden Markov model 25

2.3.2 State clustering 28

2.3.3 Triphone model 28

2.3.4 Adaptation 30

2.3.5 Maximum a posteriori estimation 30
## CONTENTS

4.2 Overview of the proposed dynamic compensation of acoustic scores . . . . 60
4.3 Acoustic score compensation . . . . . . . . . . . . . . . . . . . . . . . 61
  4.3.1 Noise and speech models . . . . . . . . . . . . . . . . . . . . . . . 61
  4.3.2 Confidence factor . . . . . . . . . . . . . . . . . . . . . . . . . . . 62
  4.3.3 Compensated acoustic scores . . . . . . . . . . . . . . . . . . . . . 63
  4.3.4 Silence likelihood alternation . . . . . . . . . . . . . . . . . . . . . 64
4.4 Experiments . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 64
  4.4.1 Setup . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 64
  4.4.2 Results . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 67
  4.4.3 Feature space techniques for dealing with noise . . . . . . . . . . . 68
  4.4.4 Distribution of confidence factors . . . . . . . . . . . . . . . . . . . 69
4.5 Further investigations . . . . . . . . . . . . . . . . . . . . . . . . . . . . 70
  4.5.1 Clustering of noise segments . . . . . . . . . . . . . . . . . . . . . 71
  4.5.2 Size of speech model . . . . . . . . . . . . . . . . . . . . . . . . . 75
4.6 Concluding remarks . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 75

5 Dynamic integration of multiple feature streams 80
  5.1 Issues on integration of multiple feature streams . . . . . . . . . . . . . . 80
  5.2 Integration of multiple streams . . . . . . . . . . . . . . . . . . . . . . . 83
    5.2.1 Stream weights based on mutual information . . . . . . . . . . . . . . 85
    5.2.2 Selective weighting (MinH) . . . . . . . . . . . . . . . . . . . . . 87
    5.2.3 Inverse entropy (InvH) . . . . . . . . . . . . . . . . . . . . . . . . 88
    5.2.4 Entropy reduction (MI) . . . . . . . . . . . . . . . . . . . . . . . . 88
  5.3 Integrated streams . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 88
    5.3.1 Energy . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 92
    5.3.2 Amplitude modulation . . . . . . . . . . . . . . . . . . . . . . . . 92
    5.3.3 Frequency modulation . . . . . . . . . . . . . . . . . . . . . . . . 93
  5.4 Experiments . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 93
    5.4.1 Setup . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 93
    5.4.2 Single stream . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 95
    5.4.3 Integrated likelihoods . . . . . . . . . . . . . . . . . . . . . . . . 97
5.5 Discussion .......................................................... 101
5.6 Concluding remarks .............................................. 103

6 Conclusion and perspectives ...................................... 105
   6.1 Contributions and conclusions ............................. 105
   6.2 Perspectives .................................................... 107
List of notations

Abbreviations

<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ASR</td>
<td>Automatic Speech Recognition (3)</td>
</tr>
<tr>
<td>NHK</td>
<td>Nippon Hoso Kyokai or Japan Broadcasting Corporation (3)</td>
</tr>
<tr>
<td>LVCSR</td>
<td>Large-Vocabulary Continuous Speech Recognition (3)</td>
</tr>
<tr>
<td>DARPA</td>
<td>Defense Advanced Research Projects Agency (5)</td>
</tr>
<tr>
<td>MLB</td>
<td>Major League Baseball (8)</td>
</tr>
<tr>
<td>HMM</td>
<td>Hidden Markov Model (19)</td>
</tr>
<tr>
<td>MFCC</td>
<td>Mel-Frequency Cepstral Coefficients (22)</td>
</tr>
<tr>
<td>DCT</td>
<td>Discrete Cosine Transform. (22)</td>
</tr>
<tr>
<td>ESA</td>
<td>Energy Separation Algorithm. (23)</td>
</tr>
<tr>
<td>TECC</td>
<td>Teager Energy Cepstral Coefficient (23)</td>
</tr>
<tr>
<td>FMP</td>
<td>Frequency Modulation Percentages (23)</td>
</tr>
<tr>
<td>CMN</td>
<td>Cepstrum Mean Normalization (24)</td>
</tr>
<tr>
<td>RASTA</td>
<td>Relative Spectral Processing (24)</td>
</tr>
<tr>
<td>FBS</td>
<td>Filter Bank Subtraction (24)</td>
</tr>
<tr>
<td>GMM</td>
<td>Gaussian Mixture Model (26)</td>
</tr>
<tr>
<td>ML</td>
<td>Maximum Likelihood (26)</td>
</tr>
<tr>
<td>EM</td>
<td>Expectation Maximization (26)</td>
</tr>
<tr>
<td>MAP</td>
<td>Maximum a Posteriori (30)</td>
</tr>
<tr>
<td>MLLR</td>
<td>Maximum Likelihood Linear Regression (31)</td>
</tr>
<tr>
<td>RTF</td>
<td>Real-Time Factor (42)</td>
</tr>
<tr>
<td>TDLM</td>
<td>Time Dependent Language Model (43)</td>
</tr>
</tbody>
</table>
WER   Word Error Rate (44)
SNR   Signal to Noise Ratio (58)
KER   Keyword Error Rate (66)
SS    Spectrum Subtraction (68)
PCA   Principal Components Analysis (80)
LDA   Linear Discriminant Analysis (81)
HDA   Heteroscedastic Discriminant Analysis (81)
MLLT  Maximum Likelihood Linear Transformation (81)
MMI   Maximum Mutual Information (81)
MCE   Minimum Classification Error (81)
ME    Maximum Entropy (81)
CMS   Comprehensive Modulation Spectra (82)
IIR   Infinite Impulse Response (88)
JND   Just Noticeable Difference (92)

**Notations**

\[ \mathbf{O} \] A sequence of observation vectors from the acoustic waveform. (18)
\[ o(n) \] An observation vector at frame \( n \). (18)
\[ \mathbf{W} \] A sequence of words from the possible linguistic representation. (18)
\[ P(\mathbf{W}|\mathbf{O}) \] A conditional probability of a sequence of words. (18)
\[ \mathbf{W}_{1:M} \] A partial string of words form \( w_1 \) to \( w_M \) (20)
\[ s(t) \] A digitized speech signal at sample \( t \). (20)
\[ \hat{s}(t) \] Pre-emphasized speech signal. (20)
\[ \text{ham}(n,t) \] An extraction window of frame data. (21)
\[ n \] A frame index. (21)
\[ T_s \] A frame shift. (21)
\[ T_w \] A frame width. (21)
\[ h(t) \] A impulse response of a vocal tract filter. (21)
\[ g(t) \] Glottal air pulses or random noise. (21)
\[ \mathcal{F} \] Fourier transform. (21)
<table>
<thead>
<tr>
<th>Notation</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\mathcal{F}^{-1}$</td>
<td>Inverse Fourier transform. (21)</td>
<td></td>
</tr>
<tr>
<td>$H(\Omega)$</td>
<td>Frequency representation of vocal tract filter response. (21)</td>
<td></td>
</tr>
<tr>
<td>$G(\Omega)$</td>
<td>Frequency representation of glottal air pulses or random noise. (21)</td>
<td></td>
</tr>
<tr>
<td>$\hat{S}$</td>
<td>A short-term power spectrum of clean speech. (24)</td>
<td></td>
</tr>
<tr>
<td>$\hat{N}$</td>
<td>An estimate of the short-term noise spectrum. (24)</td>
<td></td>
</tr>
<tr>
<td>$S$</td>
<td>A short-term power spectrum of noisy input. (24)</td>
<td></td>
</tr>
<tr>
<td>$k_{ss}$</td>
<td>The maximum allowable attenuation. (24)</td>
<td></td>
</tr>
<tr>
<td>$\alpha_{ss}, \beta_{ss}, \gamma_{ss}$</td>
<td>Scaling parameters for the spectrum subtraction. (24)</td>
<td></td>
</tr>
<tr>
<td>$q_j$</td>
<td>An HMM state (25)</td>
<td></td>
</tr>
<tr>
<td>$j = 1 : N_q$</td>
<td>A state index and a number of defined state. (25)</td>
<td></td>
</tr>
<tr>
<td>$a_{i,j}$</td>
<td>A transition probability from state $q_i$ to $q_j$. (25)</td>
<td></td>
</tr>
<tr>
<td>$b_j(o(n))$</td>
<td>An emission probability or an observation probability. (25)</td>
<td></td>
</tr>
<tr>
<td>$\mathcal{N}(\cdot)$</td>
<td>A multi-variate Gaussian distribution. (26)</td>
<td></td>
</tr>
<tr>
<td>$\mu_{jm}$</td>
<td>A mean vector of $m$th mixture component. (26)</td>
<td></td>
</tr>
<tr>
<td>$\Sigma_{jm}$</td>
<td>A covariance matrix of $m$th mixture component. (26)</td>
<td></td>
</tr>
<tr>
<td>$c_{jm}$</td>
<td>A mixture weight of $m$th mixture component. (26)</td>
<td></td>
</tr>
<tr>
<td>$\pi_j$</td>
<td>An initial probability of state $j$. (26)</td>
<td></td>
</tr>
<tr>
<td>$\theta$</td>
<td>HMM parameters consisting ${a_{ij}, b_j(\cdot), \pi_j; i, j = 1 : N_q}$ (26)</td>
<td></td>
</tr>
<tr>
<td>$X$</td>
<td>A set of observed data from the training database. (26)</td>
<td></td>
</tr>
<tr>
<td>$Y$</td>
<td>A hidden random variable used in EM algorithm. (26)</td>
<td></td>
</tr>
<tr>
<td>$Q$</td>
<td>A state sequence of training data. (27)</td>
<td></td>
</tr>
<tr>
<td>$\hat{Q}$</td>
<td>A hidden sequence. (27)</td>
<td></td>
</tr>
<tr>
<td>$\hat{\mu}_{jm}$</td>
<td>An adapted mean vector. (30)</td>
<td></td>
</tr>
<tr>
<td>$\tau$</td>
<td>An weighting factor used in MAP. (30)</td>
<td></td>
</tr>
<tr>
<td>$A_s$</td>
<td>A regression matrix estimated by MLLR (31)</td>
<td></td>
</tr>
<tr>
<td>$b_s$</td>
<td>A bias vector estimated by MLLR (31)</td>
<td></td>
</tr>
<tr>
<td>$\alpha_j^{\text{Viterbi}(n)}$</td>
<td>A Viterbi score. (31)</td>
<td></td>
</tr>
<tr>
<td>$\mathcal{L}(\cdot)$</td>
<td>Likelihood (32)</td>
<td></td>
</tr>
<tr>
<td>$\mathcal{W}^{\text{Auc}}$</td>
<td>An acoustic multiplier. (33)</td>
<td></td>
</tr>
<tr>
<td>$\mathcal{W}^{\text{lang}}$</td>
<td>A language multiplier. (33)</td>
<td></td>
</tr>
<tr>
<td>$O_{k_{\text{seg}}}$</td>
<td>An observation sequence of segment $k$. (39)</td>
<td></td>
</tr>
<tr>
<td>Notation</td>
<td>Description</td>
<td></td>
</tr>
<tr>
<td>----------</td>
<td>-------------</td>
<td></td>
</tr>
<tr>
<td>$C(O_{k}^{seg})$</td>
<td>A function assigns a segment $O_{k}^{seg}$ to a cluster $c$. (39)</td>
<td></td>
</tr>
<tr>
<td>$c_{ci}^{gmm}$</td>
<td>A mixture weight of $i$-th component of $c$-th GMM. (40)</td>
<td></td>
</tr>
<tr>
<td>$\mu_{ci}^{gmm}$</td>
<td>A mean vector of $i$-th component of $c$-th GMM. (40)</td>
<td></td>
</tr>
<tr>
<td>$\Sigma_{ci}^{gmm}$</td>
<td>A covariance matrix of $i$-th component of $c$-th GMM. (40)</td>
<td></td>
</tr>
<tr>
<td>$\theta_{c}^{gmm}$</td>
<td>A set of GMM parameters of $c$-th cluster. (40)</td>
<td></td>
</tr>
<tr>
<td>$\Delta t$</td>
<td>A length of a short fragment at the beginning of the input. (42)</td>
<td></td>
</tr>
<tr>
<td>$\lambda^{n}$</td>
<td>A set of noise models. (62)</td>
<td></td>
</tr>
<tr>
<td>$\lambda_{k}^{n}$</td>
<td>A noise model of $k$-th noise cluster. (62)</td>
<td></td>
</tr>
<tr>
<td>$N^{n}$</td>
<td>A number of the noise clusters. (62)</td>
<td></td>
</tr>
<tr>
<td>$\lambda^{s}$</td>
<td>A set of speech models. (62)</td>
<td></td>
</tr>
<tr>
<td>$\lambda_{l}^{s}$</td>
<td>A speech model of $l$-th speech cluster. (62)</td>
<td></td>
</tr>
<tr>
<td>$N^{s}$</td>
<td>A number of the speech clusters. (62)</td>
<td></td>
</tr>
<tr>
<td>$P(\lambda^{s}</td>
<td>o(n))$</td>
<td>A posterior probability of the speech model. (62)</td>
</tr>
<tr>
<td>$W^{ac}(n)$</td>
<td>A confidence factor multiplied to an acoustic score. (62)</td>
<td></td>
</tr>
<tr>
<td>$\alpha_{j}^{Viterbi}(n)$</td>
<td>A Viterbi score. (63)</td>
<td></td>
</tr>
<tr>
<td>$N^{nc}$</td>
<td>A number of noise clusters. (70)</td>
<td></td>
</tr>
<tr>
<td>$\dot{\lambda}^{s}$</td>
<td>A smaller speech model. (70)</td>
<td></td>
</tr>
<tr>
<td>$\dot{\lambda}_{k}^{n}$</td>
<td>A noise model generated automatically. (71)</td>
<td></td>
</tr>
<tr>
<td>$o_{k}(n)$</td>
<td>An observation vector of $k$-th stream. (85)</td>
<td></td>
</tr>
<tr>
<td>$N^{strm}$</td>
<td>A number of streams to be integrated. (85)</td>
<td></td>
</tr>
<tr>
<td>$I_{k}(n)$</td>
<td>An entropy reduction calculated by $k$-th stream. (85)</td>
<td></td>
</tr>
<tr>
<td>$H^{0}(n)$</td>
<td>A marginal entropy of search space. (85)</td>
<td></td>
</tr>
<tr>
<td>$\Lambda(n)$</td>
<td>A search space consisting of active states. (85)</td>
<td></td>
</tr>
<tr>
<td>$H_{k}(n)$</td>
<td>An entropy of $\Lambda(n)$ given observation vector. (85)</td>
<td></td>
</tr>
<tr>
<td>$N^{active}(n)$</td>
<td>A number of active states $\Lambda(n)$. (85)</td>
<td></td>
</tr>
<tr>
<td>$P(\lambda</td>
<td>o_{k}(n))$</td>
<td>A posterior probability of $\lambda$ given observation $o_{k}(n)$. (86)</td>
</tr>
<tr>
<td>$\bar{P}(o_{k}(n)</td>
<td>\lambda)$</td>
<td>A statically weighted observation probability. (86)</td>
</tr>
<tr>
<td>$w_{k}$</td>
<td>A static stream weight. (87)</td>
<td></td>
</tr>
<tr>
<td>$W_{k}^{strm}(n)$</td>
<td>A dynamic stream weight of $k$-th stream. (87)</td>
<td></td>
</tr>
<tr>
<td>$W_{k}^{MI}(n)$</td>
<td>A proposed dynamic stream weight based on the entropy reduction. (87)</td>
<td></td>
</tr>
<tr>
<td>$W_{k}^{MinH}(n)$</td>
<td>A dynamic stream weight of selective weighting. (87)</td>
<td></td>
</tr>
</tbody>
</table>
List of notations

$\mathcal{W}_k^{InvH}(n)$ A dynamic stream weight based on the inverse entropy. (87)
$t$ A digitized sample index. (88)
$r$ A filter index. (88)
$\mathcal{A}$ A gain parameter of a gamma-tone filter. (88)
$f_r$ A center frequency of filter $r$. (88)
$ERB(\cdot)$ A function giving equivalent rectangular bandwidth. (88)
$y_r(t)$ An output signal of $r$-th gamma-tone filter. (88)
$G^R_r(t)$ An impulse response for the real part of $r$-th gamma-tone filter. (88)
$G^I_r(t)$ An impulse response for the imaginary part of $r$-th gamma-tone filter. (88)
$A_r(t)$ A decomposed amplitude of signal $y_r(t)$. (90)
$\phi_r(t)$ A decomposed phase of signal $y_r(t)$ (90)
List of Figures

1.1 An example of a closed captioning service. ....................... 2
1.2 An example of the one seg viewer on a mobile phone ........ 3
1.3 Broadcast news captioning system ............................... 7
1.4 Live captioning for major league baseball ...................... 9
1.5 Captioning system with a re-speak method for live programs . 11
1.6 Metadata generation on live sports broadcast .................. 14

2.1 Overview of a statistical speech recognition ................... 19
2.2 Example of a Hidden Markov Model .............................. 25
2.3 An example of triphone sequence ............................... 28
2.4 An example of state clustering with a decision tree .......... 29
2.5 Overview of dynamic approaches proposed in this thesis ...... 34

3.1 Overview of the proposed dynamic selection of acoustic models. 37
3.2 An example of timescales of the databases used in two-stage clustering. 38
3.3 Outline of the first stage of the clustering ...................... 39
3.4 Outline of the second stage of the clustering .................. 41
3.5 Word error rates and real-time factors in each number of clusters. 45
3.6 Word error rates and real-time factors in each fragment length when the number of clusters was 12. .......................... 47
3.7 Word error rates in each fragment length parameterized by number of clusters. 48
3.8 Distribution of segments uttered each speaker to the clusters. .. 51
3.9 Error reduction rate in each cluster ............................ 53
3.10 Averaged GMM likelihood of each cluster. ............................................. 54
3.11 WER and RTF reduction by the speakers. ............................................. 54

4.1 Overview of the proposed dynamic compensation of acoustic scores ....... 59
4.2 Histogram of the confidence factor $W_{ac}(n)$ obtained from acoustic model
$\theta^{rn}$ shown in Table 4.2. ................................................................. 70
4.3 Recognition results of the automatically generated noise model $\hat{\theta}^n$ for 1
to 16 noise clusters. WER is plotted on the left-axis and error reductions
relative to the result of the baseline which did not compensate the acoustic
scores are plotted on the right-axis. ..................................................... 72
4.4 Recognition results of the automatically generated noise model $\hat{\theta}^n$ for 1
to 16 noise clusters. KER is plotted on the left-axis and error reductions
relative to the result of the baseline which did not compensate the acoustic
scores are plotted on the right-axis. ..................................................... 73
4.5 Recognition results of the small speech model $\hat{\lambda}^s$ for 1 to 16 noise clusters
$\hat{\lambda}^n$. WER is plotted in the left vertical axis and an error reduction relative
to the result without compensation is plotted on the right vertical axis. .... 76
4.6 Recognition results of the small speech model $\hat{\lambda}^s$ for 1 to 16 noise clusters
$\hat{\lambda}^n$. KER is plotted in the left vertical axis and an error reduction relative to
the result without compensation is plotted on the right vertical axis. .... 77

5.1 Outline of the proposed dynamic stream weighting. .............................. 84
5.2 Integrated streams. ................................................................. 89
5.3 Feature of energy calculated from Japanese speech of “m a c h i n i m a Q t a”. 90
5.4 Feature of AM calculated from Japanese speech of “m a c h i n i m a Q t a”. 91
5.5 Feature of FM calculated from Japanese speech of “m a c h i n i m a Q t a”. 91
5.6 Comparison of proposed dynamic weights $W_{k}^{MI}(n)$ and conventional dynamic weights $W_{k}^{InvH}(n)$. The top graph shows the envelope of an input waveform. The second graph shows the number of active states in the log scale together with recognition results. The WER of the result shown in E:, which is the result for “Energy” and $W_{k}^{InvH}(n)$, was 50%. The WER of the result shown in C:, which is the result of $W_{k}^{MI}(n)$, was 0%. The third and bottom graphs show $W_{k}^{MI}(n)$ and $W_{k}^{InvH}(n)$ for the same time period as the top graph.
List of Tables

3.1 Sizes of databases used in the experiments. .......................... 43
3.2 Comparisons of the proposed clustering with conventional clusterings. . 49
3.3 Number of adaptation segments. ........................................ 51
3.4 Comparison of the recognition results of regular speakers and the others. . 52
4.1 Recognition results of clean HMM $\theta^c$ with or without the proposed compensation. Relative reduction [%] is shown in parentheses. ............... 67
4.2 Recognition results of noisy HMM $\theta^n$ with or without the proposed compensation. Relative reduction [%] is shown in parentheses. ............... 67
4.3 Recognition results of $\theta^c$ and $\theta^n$ with or without the proposed compensation. Relative reduction [%] is shown in parentheses. ............... 68
5.1 Search parameters used in this experiment. .......................... 95
5.2 Recognition results of individual feature-streams. ......................... 95
5.3 Recognition results of joint feature streams. .......................... 96
5.4 Recognition results of stream integration ($\forall k, w_k = 1.0$). ............... 97
5.5 Recognition results of stream integrations using optimized static stream-weights $w_k$ of 0.51:0.28:0.21 (Energy:AM:FM). ......................... 98
5.6 Recognition results of the dynamic stream weights calculated by the whole states. ......................................................... 100
Acknowledgements

All the studies in the thesis are based on my research for eight years in human science research divisions of NHK (Japan broadcasting corporation) Science and Technical Research Laboratories and Waseda University. I would like to thank the following people for supporting me throughout this work:

Professor Tetsunori Kobayashi who was my main supervisor, and Professors Masaaki Honda, Yasuo Matsuyama, and Naohisa Komatsu who acted as vice supervisors, for giving me this opportunity, as well as for their kind suggestions and arrangements concerning the thesis presentation.

Particularly, Professor Kobayashi for his enlightening guidance and warm-hearted support. The members of his laboratory for being available, whenever I needed help. Foremost, Dr. Tetsuji Ogawa for his help and fruitful discussions.

My colleagues of NHK Sci. & Tech. Res. Labs. for their kind suggestions and discussions. Particularly, Dr. Toru Imai for this great support at NHK Sci. & Tech. Res. Labs. who spent a lot of his time to provide me with everything I needed to keep my work going. Dr. Kenkichi Tanioka (director-general of NHK Sci. & Tech. Res. Labs), Dr. Takayuki Ito (director of Human Science Division of NHK Sci. & Tech. Res. Labs), and Dr. Tohru Takagi (deputy director of Human Science Division of NHK Sci. & Tech. Res. Labs) for their great administrative support and for giving me the opportunity to this work.

My wife Noriko and children, Aoni and Sogen, for motivating me every day to keep on working. My friends who believed in me. And especially: my parents, for their patience and support.
Chapter 1

Introduction

In the broadcast raw chapter II, article 7, the public broadcaster is provided as follows:

“The purpose of NHK is to conduct its domestic broadcasting or to entrust its broadcast programs to be broadcasted with abundant and high quality broadcast programs for the public welfare and in such a manner that these broadcasting may be received all over Japan, also to conduct business necessary for the development of broadcasts and reception and at the same time to conduct international broadcasting and NHK’s international broadcast programming operations”.

It is one of the broadcaster’s pursuits to make a rich and variety of services available to anyone, including those with special needs and the elderly, through user-friendly and universally-accessible broadcasting.

The launch of digital satellite TV broadcasting in December 2000 was followed by the first digital terrestrial broadcast in December 2003. A new world of digital services are being explored. Key technologies of those digital broadcasts are Hi-Vision (high definition TV) and data broadcasting. A closed captioning is one of the service involved in the data broadcasting. The closed captions shown in Figure 1.1, which are transcriptions of spoken contents, would help viewers with hearing difficulties to follow a broadcast. With the spread of the digital-broadcasting receiver, which equips a decoder and a viewer function for the captions, the closed captions are becoming increasingly popular.
CHAPTER 1. INTRODUCTION

On the other hand, the one seg shown in Figure 1.2, is a service made possible by digital terrestrial broadcasting. It allows sharp and virtually distortion-free reception of video and audio by mobile phones, car navigation systems and other portable devices, even when in motion. As well as TV programming, the one seg offers data broadcasting of news, weather and other information, making it available on the go, on demand. Of course, the closed captioning service is also available. New viewing styles brought about such digital technology involve various combination of modalities. A combination of video and captions, for example, are preferred in a public transportation, such as a train or a bus. Thus, the closed caption is becoming a universal service and is not only for viewers with hearing difficulties.

Advances in digital technology also make viewers being able to customize their TV experience. Programs digitally broadcasted along with accompanying metadata allows us to view personalized digests such as a collection of the goals scored in soccer matches. This customization also would help viewers who are not allowed enough time to view the whole content for various reasons.
At the same time, the recent researches of automatic speech recognition (ASR), which is a technology of converting a speech signal to a sequence of words, have been promoted the technology into a practical use. The technology is expected to support wider productions of closed captions and extraction of metadata for live broadcasting. NHK (Nippon Hoso Kyokai or Japan broadcasting corporation) developed a real-time large-vocabulary continuous speech recognition (LVCSR) system for the closed-captioning service [1–4]. The developed ASR system has definitely been a success in practice with read speech, but it has not been accurate enough for the practical applications with the conditions of noisy speech, spontaneous speech, or speech uttered by unspecified speakers.

The ASR technology, likewise, is expected to assist metadata extraction for making descriptions of “who did what and when” [5]. In particular, sports programs, which have relatively high audience ratings, place significant demands for segment metadata to be included in live contents and require assistance of the ASR technology for the metadata production. The ASR system also requires robustness against background noise of such sports programs and a certain speaking style of sports commentaries for the practical use.
It is indispensable for the expansion of these practical applications that the ASR can deal with such various characteristics of input speech in real time. In an algorithm which is available for a batch processing, many model parameters, such as statistical parameters of phoneme models, can be optimized posteriorly according to input speech; however, it is difficult to optimize such an amount of parameters to improve recognition performance in the real-time processing. This thesis, therefore, presents dynamic approaches which optimize a small amount of search parameters according to time-varying and various characteristics of input speech in real time. The expansion of real-time applications are expected by the dynamic search parameters.

The social background around the broadcasting service and practical issues of the real-time ASR system for the closed captioning and metadata extraction are described in the following of this chapter.

1.1 Real-time speech recognition for closed captioning

The statistics [6] reported by the ministry of health and welfare of Japan showed that the number of persons with hearing or speech disabilities were about 346,000 in August 2001. Elderly persons over 65 years old made up 68% of the persons with the disabilities, and a close correlation between the disabilities and the age is supposed. According to an annual report [7] compiled by the cabinet office of government of Japan, as of October 1st, 2004, the number of elderly aged 65 or older reached a new high of 24.88 million and the percentage of the elderly to the total population also increased to 19.5%. The report estimated that the elderly population will continue to increase rapidly until 2020 and stabilize thereafter, and the percentage of the elderly is estimated to reach 26.0% in 2015 and 35.7% in 2050 while the total population is turning to a decrease.

There is a great need for captioned programs to help hearing impaired and elderly people to watch TV programs by displaying captions of spoken words. The captions can be created by manually transcribing and summarizing spoken contents of the programs apart from live broadcasting. In 2006, NHK manually created captions 98.2% of the programs of which production has been completed prior to the broadcasting.
On the other hand, simultaneous subtitling of live broadcasting, particularly news programs, is in high demand though the captions of such live programs must be produced in real-time. Although a special keyboard can be used for real-time captioning, there are restrictions on the caption production using the keyboard due to the special skill required for keyboard operators and their operation costs. NHK, therefore, has undertaken extensive research in automatic speech recognition for captioning live news programs in real-time.

In the U.S., continuous speech recognition for a large vocabulary has been studied centering on HUB4 of defense advanced research projects agency (DARPA) [8] with regarding speech recognition for news programs. The main studies of the project transcribing broadcast news were different from ours in following points.

1. They were based on batch processing and did not address real-time processing.

2. They focus on information retrieval by employing the result of speech recognition rather than accurate conversion of spoken words into characters.

The studies were, therefore, different from real-time speech recognition which instantly produces captions for live news programs.

From March 2000 to March 2006, NHK broadcast subtitled news every day by using a Japanese continuous speech recognition system we developed [1, 2]. The captioning was limited to portions of programs where an anchorperson reads in a quiet studio. However ours was world’s first implementation of simultaneous captioning with speech recognition.

Besides news programs, captions of other live programs, such as music or sports programs, would also be helpful to the viewers. The commentary of such a program is spoken spontaneously and emotionally, and sometimes, speakers speak simultaneously. It is very difficult for the current speech recognition technology to recognize such commentary with a sufficient degree of accuracy. The re-speak method, therefore, is used for the captioning. In the re-speak method, a speaker listening to the original speech of the programs rephrases the commentary so it can be recognized for subtitling [9].

This section describes current status of the simultaneous captioning and issues on the speech recognition system.
1.1.1 Direct method

Broadcast news captioning

NHK started simultaneous captioning for the evening live news program “News 7” on March 27th, 2000. The captioning was later extended to “News 9” in the evening, “News at Noon”, and “Good morning Japan” in the morning. The speech recognition was used for the portions where the anchorperson reads in a quiet studio, and captions in other portions, such as field reports and spontaneous conversations, were given by using a special keyboard or prepared electronic scripts.

Figure 1.3 shows the system of simultaneous broadcast news captioning. Typical news scripts covered by reporters are electronically gathered in a database. Such news scripts are printed out and then are revised manually by chief editors or the anchorperson until just before broadcasting. These corrections often include partial deletion to adjust the time in the program and editing to integrate multiple scripts, and finally, announcers correct so that they can easily read them. In addition, some late-breaking articles are also delivered in hand written forms or through facsimile without electric processing. Because the news script to be read is not finalized until just before broadcasting, speech recognition is required for the captioning in spite of news scripts are electrically handled by the database.

The news script in the database, however, contains many useful word sequence for the speech recognition, for example a name of a person or a place name included in news topics. The speech recognition system achieves a high recognition performance by training the system using these scripts.

Speeches picked up by a microphone directed to the anchorperson in a quiet studio are directly fed into the speech recognition. Background noise in a VTR or field reporting and music are mixed separately for sound tracks. The speech is transcribed into Japanese Kanji characters in real-time. In order to output recognition results for the captions as quickly as possible, the search engine (decoder) makes an early decision without waiting for the end of input utterances [10].

In speech recognition, it is extremely difficult to achieve a recognition rate of 100%. Consequently, it is essential for humans to correct the errors in the recognition result in order to implement error-free simultaneous captioning. News programs are particularly
Figure 1.3: Broadcast news captioning system
expected to be captioned truthfully; therefore, the error correction is done in real time by four persons listening to the news. In order to make the delay as short as possible, we have two correction sets operated in a sentence by sentence which is automatically segmented by pauses in the speech. Each set has two operators: one operator detects and points out incorrect words with a touch-panel and the other one corrects the identified words with a keyboard. A speech rate conversion system [11] is employed to synchronize the speech presentation with the recognition results.

The corrected texts are encoded into the television signals and transmitted to viewers. A receiver superimposes the captions over the contents if a viewer wishes them. Typically the captions are 15 characters per line, and two lines are displayed at the bottom of the screen every four seconds.

The word accuracy of the speech recognition for the anchor-person’s prepared speech in the studio is on average 98% with a delay of 1 to 2 seconds. It takes 4 to 5 seconds for the operators to confirm and correct the results. The accuracy after correction exceeds 99%. The remaining errors are particles or trivial mistakes that have less chance to affect the meaning conveyed to the viewer. Since the captioned page is updated every 4 seconds, the delay from the moment the speech uttered is approximately 10 seconds.

**Live captioning for major league baseball**

The method of the direct speech recognition can be applied to a live sports program delivered from foreign country. Figure 1.4 shows a simultaneous captioning system for major league baseball (MLB) using a direct recognition. In this situation, live contents are delivered from a stadium in U.S. without any Japanese commentary. A Japanese running commentary on the game are inserted at a quiet studio of a broadcast-center in Japan. Noise-free utterances of a sportscaster can be transcribed by the speech recognition system.

Utterances of a guest speaker are not captioned because of their spontaneous speaking style degrading recognition accuracy. Thus, a speaker to be recognized can be specified to the sportscaster. The speaker dependent speech recognition system, therefore, can be adopted, so that the recognition system produces more accurate captions than the speaker independent system used for the simultaneous news captioning. The speaker dependent system for the sportscaster is trained in advance from utterances of the caster. The training
Figure 1.4: Live captioning for major league baseball
utterances are gathered from a speech database or records of the caster at the games before. Game specific words are also trained for each game in advance to improve recognition performance. A text database including player’s names, baseball terms, and recent sports news is used for the training. Compared with the news captioning (Section 1.1.1), the correction system is reduced in order to minimize the delay between a play and a caption, so that recognition errors are just deleted by an operator.

The captions transcribing the speech uttered by the guest speaker are also demanded because the guest speaker explains important plays and sometimes gives historical backgrounds of a player or a record. The re-speak method described in next section achieves simultaneous captioning for these spontaneous speech.

1.1.2 Re-speak method

The commentaries and conversations in live TV programs of music or sports are often spontaneous and emotional, and sometimes different speakers speak at the same time. If such utterances are recognized directly, output results of the speech recognizer will not be accurate enough for captioning. The reasons are background noise, unspecified speakers, and speaking style that may not match the text corpus or the speech data used for the training of the recognition system. It is difficult to collect enough training data in the same domain as the target program because the hand transcriptions required for the training are very expensive. Therefore, we employ the re-speak method to eliminate such difficulties.

In the re-speak method, a different speaker from the original speakers of the target program carefully rephrases what he or she hears. We call him or her the “re-speaker”. As shown in Figure 1.5, the re-speaker wearing headphones listens to the original soundtrack of live TV programs, and he re-speak what he hears or paraphrases it, if needed, so that its meaning will be clearer or more acceptable than the originals and the expression will be more easily recognized. The method has the following advantages for speech recognition. Followings are acoustical advantages of the re-speak method.

- As described in the MLB captioning Section 1.1.1, the re-spoken utterances have no background noise because they does not re-speak on site but utters in a quiet studio.
- The speaker dependent adaptation beforehand improve recognition performance by
1.1. REAL-TIME SPEECH RECOGNITION FOR CLOSED CAPTIONING

Figure 1.5: Captioning system with a re-speak method for live programs
using relatively many adaptation data.

- Since only one re-speaker inputs the speech of all speakers in a program, the speech does not overlap.

- The re-speaker does not speak emotionally but clearly and calmly without repeating filled pauses and hesitations in the original speech.

- If a recognition error happens, the re-speaker inputs the same phrase again or tries a different phrase.

- The re-speaker can supplement the speech by mentioning the sounds of audiences, such as applause, even if no mention is given in the original narrations.

The above acoustic advantages make the recognition accuracy better and the captions easier to be understood by hearing impaired viewers. Linguistic advantages of the re-speak method are follows.

- The method makes it possible to summarize or rephrase the original narrations.

- Conversational speech is rephrased into a planned speech style.

- Subjects or positional particles in an incomplete sentence are supplemented.

- Inverted sentences and difficult words or phrases are avoided.

They reduce the mismatch between the corpus used for the system training and the original contents, and make the captions more accurate and more understandable. Apparently, the way of re-speaking affects the speech recognition performance. It is necessary to have skillful re-speakers so that the final captions will be as good as possible.

Since December 2001, NHK has been using the re-speak method for automatic speech recognition and captioning of live music shows and sports events. For example, this method of subtitling was used in the Olympic games, the world cup football games, the grand sumo tournaments, and professional baseball games. The recognizers are optimized for each program and for each re-speaker. The recognition accuracy is approximately 95%, and any recognition error is promptly corrected manually. Captions can be presented within 5
to 8 seconds. A large number of positive responses from viewers about the simultaneous captioning were heard. Hearing-impaired viewers expressed delight at finally being able to enjoy programs together with their families.

It is considered that the advantages described this section are the issues to be addressed for the development of a full automatic captioning system.

1.2 Speech recognition for segment metadata

The speech recognition is hoped to facilitate new ways of viewing TV for future broadcasting based on home servers [12]. For instance, the system is desired to make it possible for a viewer to see video clips of Shunsuke’s past goal while watching a live broadcast of world cup football game. Program related-information called metadata enable a wide range of new viewing styles, such as highlight viewing [13]. Metadata is index data for TV programs and related materials. A programs’ title, its credits, and its summary are also metadata. Metadata also includes a description of individual scene, such as “a scene in which someone is doing something.” Advances are being made on ways of generating efficient metadata using speech recognition technology.

A metadata editor incorporating various analysis functions is developed to extract metadata from a program and related materials as automatically as possible [5, 14, 15]. For example, to assist with final manual editing, it extracts information about “who is doing what” in an image, and determines what is being said based on audio data. Figure 1.6 shows an example in which the metadata generation system adds various pieces of metadata to a live football broadcast. The system integrates image analysis, sound analysis, speech recognition, and language processing.

Speech recognition plays a crucial role in such system because recognized keyword can be used for a metadata straightforwardly through the language processing. The recognizer is tuned to the target program to reduce recognition errors stemming from background noise or a certain speaking style of sports commentaries. Recognition accuracy is nevertheless lower than the captioning system described in previous sections. Currently, about 80% of the keywords uttered by the announcer can be correctly extracted from the audio. Enhancement of the noise robustness is, however, required for the metadata production because
Figure 1.6: Metadata generation on live sports broadcast

Part of metadata

Meta-data editor

Broadcasting service

(who did what when)

Meta-data

Figure 1.6: Metadata generation on live sports broadcast

Part of metadata

Meta-data editor

Broadcasting service

(who did what when)

Meta-data
1.3. ORGANIZATION OF THIS THESIS

important keywords like “Goal !!” are often accompanied by loud crowd noise, which can be a clear indicator of scenes with significant events [5].

1.3 Organization of this thesis

This thesis presents dynamic approaches acoustically improving recognition performance on the issues described in this chapter.

This thesis is organized as follows.

Chapter 2, Technical introduction, describes the technical backgrounds of the large-vocabulary continuous speech recognition (LVCSR). Following acoustical components comprising the LVCSR are reviewed to explain settings of the proposed dynamic approaches.

Firstly, feature extraction techniques are described. This component projects the speech signal to a compact feature space so that the sequence of the features can be associated effectively with a linguistic unit such as a phoneme or a word.

Second component is an acoustic model comprising of statistics of features observed on each phonetic unit. An effective training algorithm of the acoustic model and a statistical method of identifying input features with words are described.

Finally, the conventional search algorithm is described. The algorithm estimates the most probable word sequence from among the hypotheses generated by using a language model. The language model provides likelihoods of given linguistic representations, and the acoustic likelihoods of word sequences giving higher linguistic likelihoods are evaluated by using the acoustic model. Successful techniques for the real-time processing are also described.

Furthermore, this chapter illustrates the association of the proposed dynamic approaches with these components.

Chapter 3, Dynamic approach selecting an acoustic model, presents a dynamic approach improving the recognition accuracy of the broadcasting contents, in which the recognition system can not specify speakers or speech environments. The method selects the most appropriate acoustic model for an input utterance in real-time from among cluster dependent acoustic models. These acoustic models are adapted in advance to the clusters of the speakers or the speech environments. In order to select the acoustic model in real-time,
the method utilizes a 0.5 seconds of speech fragment at the beginning of an input utterance and its likelihoods of the cluster models.

This chapter also addresses an efficient training method of the cluster models and the cluster dependent acoustic models. This method obtains the adaptation data for the program specific acoustic models by clustering wide varieties of speakers or speech environments of a large-scale speech database in two stage.

Transcription experiments implementing the proposed method for Japanese broadcast news were performed for the purpose of the automatic captioning. The results showed improvements in the recognition performances of accuracy and processing costs. It was remarkable that the method improved the recognition performance of reporter’s speech because they are stringy demanded to reduce the mismatch between an input and the models.

Chapter 4, *Dynamic compensation of acoustic scores*, addresses weights of the acoustic likelihoods against the linguistic likelihoods. In the proposed method, the acoustic likelihoods are dynamically weighted with time-varying reliabilities. The reliabilities are estimated from amounts of the mismatches between the training database and input speeches. In order to estimate the mismatches caused by abrupt background noise, a frame-wise confidence factor is calculated by using noise models incorporating knowledge of various kinds of noise. The noise model captures the spectral shape and its dynamics of each noise category, so that the method is able to estimate the unreliability caused by the noise even if it changing rapidly. Furthermore, the method utilizes the likelihoods of the noise models for silence parts of hypotheses, so that the method reduces the word errors caused by the noise fragments being recognized as phonemes.

Transcription experiments implementing this method for broadcast sports commentary were performed for the purpose of the captioning and the metadata extraction. The results showed lower word error rates by the proposed method, especially the recognition errors in keywords, which are important for the metadata extraction, were significantly reduced.

This chapter also proposes a method of generating the noise models by using an automatic segment clustering. The noise models generated by this method showed better results than those trained with hand-labeled noise clusters. It is concluded that the improvement given by the method should be of great advantage to the metadata extraction.

Chapter 5, *Dynamic integration of multiple feature streams*, addresses an integration
of the feature streams representing different acoustic aspects. The presented novel method dynamically integrates the likelihoods of multiple feature streams for robust speech recognition. The integration algorithm calculates a frame-wise stream weight so that a heavier weight is given to a stream that is robust to a variety of noisy environments or speaking styles. Such a robust stream is expected to bring out discriminative ability. The method utilizes the mutual information, which is calculated as the entropy reduction caused by an observed feature, to estimate the discriminative ability. The stream weight is, therefore, calculated in real time from mutual information between an input stream and active search space.

In this study, the modulation feature streams extracted through auditory filters are integrated by taking into account the human auditory system extracting amplitude and frequency modulations. These features are expected to provide complementary clues for the speech recognition.

Speech recognition experiments using field reports and spontaneous commentary from Japanese broadcast news were performed for the purpose of the captioning. The word error rates in both conditions were reduced as the results of the proposed integration.

The effectiveness of the method is also discussed with comparing to the conventional static integrations based on the discriminative analysis.

Chapter 6, Conclusion summarizes this thesis and concludes with some remarks.
Chapter 2

Technical introduction

2.1 Overview of a statistical speech recognition

This chapter reviews the most popular automatic speech recognition (ASR) using probabilistic models. The goal of the ASR system is to deduce meaningful linguistic unit, that is to say words, from acoustic waveforms. A deterministic approach that provides a mapping between acoustic signal and conceptual meanings was not successful because of the random nature of the process and interferences, but recent researches on the probabilistic framework, which has the ability to handle the uncertainty of the speech signal, achieve the practical use of the technology instead.

A general framework for the statistical ASR system can be schematically represented by Figure 2.1. As mentioned above, the probabilistic speech recognition can be states as the estimation of the most probable linguistic representation of a given acoustic waveform. The mathematical formulation of providing the probability of a word sequence can be written as

\[
\hat{W} = \arg \max_W P(W|O).
\]  

(2.1)

In this formulation, \(O\) is a sequence of observation vectors \(o(n)\) and an observation vector \(o(n)\) is a component relevant to the speech content extracted form the input speech by analyzing temporal waveform. Section 2.2 gives further explanation. \(W\) is a random variable that takes sequences of words form the possible linguistic representation and \(P(W|O)\)
is the conditional probability of the linguistic representation $W$ of given observations sequence $O$. This conditional probability is provided by the acoustic models and the language models by simplifying the complexity of $P(W|O)$ involving both acoustic and linguistic information. They can be handled separately by using a Bayesian reformulation:

$$W = \arg \max_w P(O|W)P(W). \quad (2.2)$$

In this formulation, $P(O|W)$ is provided by the acoustic models which encodes the statistical distribution of speech acoustics given the linguistic labeling. $P(W)$ is the probability assigned by the language model which encodes a priori linguistic information.

These models can be constructed by using statistical learning techniques. The information sources for the acoustic component are the speech utterances labeled with corresponding linguistic units of words, that is to say, sub-words or phonemes. State-of-art ASR systems are based on probabilistic modeling of the speech signal. Hidden Markov models (HMM) described in Section 2.3 are the most successful probabilistic modeling
of the speech signal. For the linguistic component, the information sources are written text documents which are supposed to be sufficient to represent the properties of a language, a domain, or a topic. N-gram language models [16] are commonly used in HMM-based speech recognition systems. The probability of a random word sequence $W_{1:M} = \{w_m; m = 1 : M\}$ can be written as

$$P(W_{1:M}) = P(w_1) \prod_{m=2}^{M} P(w_m|W_{1:m-1}),$$

(2.3)

where $W_{1:M}$ is the partial string of words from $w_1$ to $w_M$. Since the space of sequences is infinite, the probability is simplified and computed as a product of local probabilities over equivalent neighborhood or history $W_{m-N_{gram}+1:m-1}$.

$$P(W_{1:M}) = P(w_1) \prod_{m=2}^{M} P(w_m|W_{m-N_{gram}+1:m-1}).$$

(2.4)

The most commonly used forms of N-grams are trigram ($N_{gram} = 3$), bigram ($N_{gram} = 2$) and unigram ($N_{gram} = 1$) language models.

### 2.2 Speech analysis

The goal of the speech analysis is to attain a projection of speech signal to a compact parameter space in order to associate the sequence of the parameter with the linguistic unit by the HMMs. The first stage of this analysis obtains digitized speech signal $s(t)$ for every sample index $t$. The data rate of the digitized waveform then is reduced by converting it into frame-wise feature vectors, being considered as a quasi-stationary process.

Typically, input speech are pre-emphasized using following forms after digitized at 16kHz and 16bits,

$$\hat{s}(t) = s(t) - 0.97s(t - 1),$$

(2.5)

where $\hat{s}(t)$ is a pre-emphasized speech signal. The feature vectors are then computed every 10ms form the pre-emphasized short segments extracted with overlapping Hamming
windows of 25ms. The extraction window $\text{ham}(n, t)$ can be defined as follows,

$$
\text{ham}(n, t) = \begin{cases} 
0.54 - 0.46 \cos \left( \frac{2\pi (t - n T_s)}{Tw - 1} \right) & \text{if } n \cdot T_s < t < n \cdot T_s + T_w \\
0.0 & \text{otherwise}
\end{cases} 
$$

(2.6)

where $n$ is a frame index, $T_s$ is a frame shift of 10ms, and $T_w$ is a frame width of 25ms.

The feature vectors are extracted so as to minimize extraneous variation in the speech input. These variation may be due to differences in pitch between speakers, differences in length of the vocal tract across speakers or the nature of channel through.

### 2.2.1 Cepstrum

At present, most ASR systems are based on Cepstrum [17] parameters. The parameterization scheme is based on the source-filter model of speech production mechanism. In this model, speech signal is considered as the output of a vocal tract filter $h(t)$ whose input source is either glottal air pulses or random noise $g(t)$. For voiced sounds the glottal excitation is considered as a slowly varying periodic signal. This signal can be considered as the output of a glottal pulse filter feed with periodic impulse train. For unvoiced signal the excitation signal is considered as random noise. The characteristics of the filter $h(t)$ are related to the shape of the vocal tract and very less across example of unit of speech, while glottal component varies widely with pitch. Assuming non-zero spectrum, it is possible to separate the contribution of the two components by transforming the input speech signal as follows,

$$
\mathcal{F} [h(t) \otimes g(t)] = H(\Omega)G(\Omega) 
$$

(2.7)

$$
\log |\mathcal{F} [h(t) \otimes g(t)]| = \log |H(\Omega)| + \log |G(\Omega)| 
$$

(2.8)

$$
\mathcal{F}^{-1} [\log |\mathcal{F} [h(t) \otimes g(t)]|] = \mathcal{F}^{-1} [\log |H(\Omega)|] + \mathcal{F}^{-1} [\log |G(\Omega)|] 
$$

(2.9)

where, $\mathcal{F}$ and $\mathcal{F}^{-1}$ denote the Fourier transform and its inverse [18]. $H(\Omega)$ and $G(\Omega)$ are frequency representation of $h(t)$ and $g(t)$ respectively.

The glottal pulse has harmonics related to the pitch of the speaker and appears much higher in the cepstral domain compared to the response of the vocal tract. By computing
the lower (12-14) order cepstral coefficients, the variation due to pitch is largely left out while retaining most of the essential information for speech recognition.

**MFCC**

Motivated by the robustness of human auditory system, certain transformations are applied in the frequency domain before computing the cepstral coefficients. In state of the art speech feature extraction schemes of mel frequency cepstral coefficients (MFCC) [19], the frequencies are warped to resemble human auditory processing by using the mel scale, almost constant Q-filters defined as follow,

\[
Mel(f) = 2595 \cdot \log_{10} \left(1 + \frac{f}{700}\right).
\]  

(2.10)

The magnitude or power of each frequency bin are combined into 24 mel-spaced bins by using weights being triangular in shape for each mel-spaced bin typically. The log-spectral value corresponding to combined value of mel-spaced bin are then obtained.

An inverse discrete cosine transform (DCT) is used to perform the inverse Fourier transform described in (2.9) so as to obtain de-correlated coefficients. After applying these techniques, input speech is represented typically in the form of 12-14 cepstrum coefficients. These cepstral features, their derivatives, and the log of local energy are concatenated together to form a feature vector. A sequence of these vectors at a rate of about 100 par second are the input to the recognition system.

### 2.2.2 Short-term modulation

The spectrum oriented feature described above makes the assumption that resonances, that is to say, the center frequency of the formants, remain constant in an analysis frame. The speech resonances, however, have dynamic nature, so that they can fluctuate around their center frequency and can be modeled as the sums of AM-FM signals [20]. It is reported that these fluctuations give auxiliary clues for human speech perception in noisy environment [21]. Short-term modulation features quantifying these fluctuations can be used for speech recognition and may improve recognition performance [22]. These features are also
expected to provide different clue for the cepstrum-based features complementary.

In the literature, several kinds of demodulation algorithms have been proposed to obtain instantaneous amplitude and frequency from band-passed speech signals. The energy separation algorithm (ESA) [20] based on the Teager-Kaiser energy operator is one of the most successful approach of extracting these signals. Approaches using the Kalman filtering [23] and the Hilbert transform [24] also have been succeeded in extracting such signals.

The Teager energy cepstral coefficient (TECC) [25] is one of the feature taking into account of the short-term modulation. The feature replace the square amplitude of the band-passed speech signal by the nonlinear Teager energy in the standard estimation algorithm of the MFCC feature set.

The frequency modulation percentages (FMP) [26,27] captures the fluctuation directly. The features are the ratios of the second over the first moments of instantaneous frequencies extracted by using the ESA. The spectral moments joined with the conventional spectrum oriented features have been tested in various ASR tasks and yielded improved results.

In a similar manner, features taking into account of amplitude modulation can also be extracted by using their statics, that is to say, their first and second spectral moments of instantaneous amplitude extracted by using the ESA. It has been shown that they contain significant amount of information concerning both the speaker and the linguistic content of the speech signal [28].

2.2.3 Long-term modulation

The short-term modulations described above are studied in time-windows up to 10-30ms in order to capture the micro-details (very rapid changes) of the speech signals. On the contrary, long-term modulations are examined in order to capture the temporal evolution of the spectral energy. In this analysis, the corresponding time-windows are in the range of 200-500m seconds.

The literature on the perceptual ability on the human auditory system have shown that the speech intelligibility is not affected by low-pass filtering below 16Hz, or high-pass filtering above 4Hz [29, 30]. In noisy environment, the modulation frequency between 2 and 8Hz plays important role for the speech intelligibility [31].
The regression coefficients [32], that is to say delta and acceleration coefficients, attempt to incorporate such long-term temporal information into the recognition feature. These coefficients are computed by first- and second-order orthogonal polynomial expansions of the time trajectories of the feature. The method of cepstrum mean normalization (CMN) subtracts the long-term average from logarithmic speech spectrum and suppresses convolutive noise. Similarly, relative spectral processing (RASTA) [33, 34] suppresses the modulation frequency components that do not belong to the range from 1 to 12Hz. Thus, this method suppresses the slowly varying convolutive distortions and attenuates the spectral components that vary more rapidly than the typical rate of change of speech.

2.2.4 Spectral subtraction

Spectral subtraction [35] is a technique of suppressing additive noise on the contrary to the previous method of suppressing convolutive noise. This method estimate a short-term power spectrum of clean speech \( \hat{S} \) by subtracting an estimate of the short-term noise spectrum \( \hat{N} \) from noisy input spectrum \( S \). Explicitly,

\[
\hat{S} = \begin{cases} 
\left( \frac{|S|^\gamma_{ss} - \alpha_{ss} |\hat{N}|^\gamma_{ss}}{|S|^\gamma_{ss}} \right)^{\beta_{ss}} S, & \text{if } |S|^\gamma_{ss} - \alpha_{ss} |\hat{N}|^\gamma_{ss} > k_{ss} |S|^\gamma_{ss} \\
\frac{k_{ss}}{S} & \text{otherwise}
\end{cases}
\]

(2.11)

where \( k_{ss} \) is the maximum allowable attenuation of the corrupted signal and a set of parameters \( \alpha_{ss}, \beta_{ss}, \gamma_{ss} \) are the scaling parameters. Many optimization method of these four valuables are proposed for various applications.

For the MFCC based speech recognition, filter bank subtraction (FBS) [36] is proposed to improve the recognition performance of the field reporting of broadcast news programs. This method precisely estimates the instantaneous power distribution of the background noise \( \hat{N} \) by finding the minimum component in mel-spaced frequency bins and robustly subtracts them in mel-frequency domain.
2.3. ACOUSTIC MODEL

Figure 2.2: Example of a Hidden Markov Model

2.3 Acoustic model

2.3.1 Hidden Markov model

As described in Section 2.1, the acoustic model provides observation probabilities of $P(O|W)$ given word sequence $W$. The hidden Markov model (HMM) is the most popular and successful stochastic approach to the speech recognition [37–39]. In the LVCSR, every words in the vocabulary are partitioned into sub-word units modeled by using the hidden Markov model. Therefore, a sequence of words, that is to say, an utterance or a sentence, are represented by a concatenation of the HMMs corresponding to the sub-word sequence.

Figure 2.2 shows an example of a hidden Markov model representing a sub-word. It consists of a collection of states $\{ q_j; j = 1 : N_q \}$. At a discrete time step one can make transition from state $q_i$ to state $q_j$ with a transition probability $a_{i,j}$. When any emitting state $q_j$ is entered at frame $n$, a feature vector $o(n)$ is emitted, based on an underlying distribution $b_j(o(n)) = P(o(n)|q_j)$. The emission probabilities $b_j(o(n))$ are modeled as multivariate Gaussian densities or a mixture of multivariate Gaussian densities. For the
Gaussian mixture model (GMM) case, the probability density function is given by

$$P(o(n)|q_j) = \sum_{m=1}^{M} c_{jm} \mathcal{N}(o(n); \mu_{jm}, \Sigma_{jm})$$  \hspace{1cm} (2.12)

where $\mathcal{N}(\cdot)$ is a multi-variate Gaussian distribution with mean vector $\mu_{jm}$ and covariance matrix $\Sigma_{jm}$ of a mixture component $m$ and each mixture component has an associated weight $c_{jm}$. Furthermore, an initial probabilities of each state is given by \{ $\pi_j = P(q_j); j = 1 : N_q$ \}.

In the mainstream LVCSR, these stochastic model parameters are automatically estimated from a large speech database. There are many criteria that may be chosen to optimize these model parameters, however, the most popular criterion is the maximum likelihood (ML) mainly due to the existence of very efficient training algorithms. It is also optimal when the form of the model is correct. Although there is no closed form solution to estimate HMM parameters $\theta = \{a_{ij}, b_j(\cdot), \pi_j; i, j = 1 : N_q \}$ from a training set, we can choose an initial $\theta$ and iteratively improve the estimate of $\theta$ using the Baum-Welch algorithm, which is an instance of expectation-maximization (EM) algorithm [38]. The EM algorithm iteratively maximize a particular cost function $L(\theta|\theta_{old})$ which guarantees to increase the likelihood of data given a model $P(X|\theta)$, where $X$ is a set of observed data from the training database. The cost function $P(\theta|\theta_{old})$ is defined in terms of a hidden random variable $Y$.

$$L(\theta|\theta_{old}) = E[\log P(X, Y|\theta)|X, \theta_{old}]$$

$$= \sum_{Y} P(Y|\theta_{old}) \log P(X, Y|\theta)$$  \hspace{1cm} (2.13)

$$L(\theta_{new}|\theta_{old}) \geq L(\theta_{old}|\theta_{old}) \implies P(X|\theta_{new}) \geq P(X|\theta_{old})$$  \hspace{1cm} (2.14)

The unobserved random variable $Y$ is carefully chosen for each application so that the computation of $L(\theta|\theta_{old})$ is simplified. In general, the iterations converge to a local maxima.

In the case of an HMM, the computation of $L(\theta|\theta_{old})$ is simplified by considering the
hidden state sequence \( Q = \{q^m; m = 1 : M\} \) as the variable \( Y \) and the training observation sequence \( \{O = \{o(n); n = 1 : N_o\} \) as the variable \( X \) in Equation 2.13, where \( M \) is the number of the states and \( N_o \) is the number of observation frames involved in the training data. Then, the expected joint log probability over hidden sequence \( \hat{Q} = \{\hat{q}(n); n = 1 : N_o\} \) can be computed in terms of \( P(\hat{q}(n)|O, \theta_{old}) \) and \( P(\hat{q}(n - 1), \hat{q}(n)|O, \theta_{old}) \) thorough two recursions, since

\[
E [\log P(O, Q|\theta)|O, \theta_{old}] = \sum_Q P(\hat{Q}|O, \theta_{old}) \log P(O, \hat{Q}|\theta) \tag{2.15}
\]

\[
= \sum_j P(q(1) = j|O, \theta_{old}) \log \pi_j
+ \sum_n \sum_j \sum_k P(q(n - 1) = j, q(n) = k|O, \theta_{old}) \log a_{jk}
+ \sum_n \sum_j P(q(n) = j|O, \theta_{old}) \log b_j(o(n)) \tag{2.16}
\]

The posterior probability of being a state \( P(q(n) = j|O, \theta_{old}) \) and the transition probability \( P(q(n - 1) = j, q(n) = k|O, \theta_{old}) \) can be computed efficiently using forward-backward algorithm as shown below.

\[
\gamma_j(n) \triangleq P(q(n) = j|O, \theta_{old}) = \frac{P(q(n) = j, O|\theta_{old})}{\sum_j P(q(n) = j, O|\theta_{old})} \tag{2.17}
\]

\[
P(q(n) = j, O|\theta_{old}) = P(q(n) = j, O_{1:n}|\theta_{old})P(O_{n+1:N_o}|q(n) = j, \theta_{old}) \tag{2.18}
\]

where \( O_{n:m} \) is a partial sequence of \( O \) from \( n \) to \( m \). So, the key quantities to compute the expectation step in the EM algorithm are,

\[
\alpha_i(n) \triangleq P(q(n) = i, O_{1:n}|\theta_{old}) = b_i(o(n)) \sum_{j=1}^{N_q} \alpha_j(n - 1)a_{ji} \tag{2.19}
\]

\[
\beta_i(n) \triangleq P(O_{n+1:N_o}|q(n) = i, \theta_{old}) = \sum_{j=1}^{N_q} \beta_j(n + 1)b_j(n + 1)a_{ij} \tag{2.20}
\]
CHAPTER 2. TECHNICAL INTRODUCTION

Figure 2.3: An example of triphone sequence

\[
\gamma_i(n) = \frac{\alpha_i(n)\beta_i(n)}{\sum_j \alpha_j(n)\beta_j(n)} \quad (2.21)
\]

\[
\xi_{ij}(n) \triangleq P(q(n-1) = i, q(t) = j|O, \theta_{old}) = \frac{\gamma_{ij}(n)a_{ij}b_{j}(n+1)\beta_j(n+1)}{\beta_i(n)} \quad (2.22)
\]

The \( \alpha \) and \( \beta \) recursions in Equation (2.19),(2.20) are computed in forward and backward direction respectively after they are initialized with \( \alpha_i(1) = \beta_i(N_q) = 1 \). In each iteration, the HMM parameters are updated using sufficient statistics gathered in that iteration.

2.3.2 State clustering

Triphone model

The acoustic nature of a phoneme is highly dependent on the preceding and following phonemes. The variants of a phoneme are called allophones. A typical LVCSR system takes these variants into account by modeling words as a sequence of context dependent phonemes. Triphones are the most popular sub-word units where each triphone is defined as a phone with a unique pair of left and right context (Figure 2.3). These context-dependent models make it difficult to obtain sufficient data required for the accurate estimation of model parameters for all the triphones observed in the training set. Furthermore, the training data may not cover all the triphones appeared in the test set. A widely accepted solution to this problem allows tying sets of HMM parameters together. The tying may be applied at many levels, ranging from the model level to the Gaussian component level. The most
2.3. ACOUSTIC MODEL

The commonly adopted approach shares parameters of the observation density across states of different sub-word units. This state-tying structure can be determined automatically using the linguistic class of the associated labels in decision trees [40] [41] [42]. The decision trees are trained by using training data. At first, all the triphone observed in the training data are pooled at the root node of the tree, as shown in Figure 2.4. A set of predefined questions, e.g., Right:Voiced (R:Voic. groups the triphones whose right context are voiced phoneme), is used to define partitions of node in the tree. The question corresponding to a partition which maximizes the likelihood of the training data assigned to a node is chosen as a candidate for the next split. The best partitions of the new clusters resulting from this split procedure are added to the list of the candidate splits, and thus tree is grown until some stopping criterion is met.
It is noted that the state-tying structures are dependent on the training data which are typically the same one used for estimation of the HMMs parameters. The sub-word network, which is a merged and rearranged lexicon in a tree form, can be shrunken by merging these tied states. This tree formed lexicon improves the efficiency of hypothesis search of the large vocabulary speech recognizer [43].

### 2.3.3 Adaptation

The recognition performance is degraded, often dramatically, if there is some mismatch between the training and test acoustic environment such as a particular speaker or a particular noise environment. For the speech recognition systems, a large amount of data is required to retain the system for such a particular speaker or a new acoustic environment. Adaptation techniques are able to improve the performance of an existing system while only using a small amount of speaker-specific or environment-specific adaptation data.

#### Maximum a posteriori estimation

The maximum a posteriori (MAP) estimation is one of the basic adaptation methods [44, 45]. It uses prior information about the parameters of existing models. This usage of prior information in this training process solves the problem caused by sparse training data which leads to inaccurate estimates in standard maximum likelihood (ML) estimation.

The solution of MAP estimation for mean is:

\[
\hat{\mu}_{jm} = \frac{\tau \mu_{jm} + \sum_n \gamma_{jm}(n) o(n)}{\tau + \sum_n \gamma_{jm}(n)}
\]  

(2.23)

where \(\hat{\mu}_{jm}\) is an adapted mean vector and \(\tau\) is the weighting factor which determines convergence speed. The re-estimation formula for mean vector represents a weighted sum of the prior mean vector and standard ML estimate. When the amount of adaptation data increase, then \(\lim_{n \to \infty} \sum_n \gamma_{jm} = \infty\) and the MAP estimation and the ML estimation give identical results. The MAP estimates for variances and mixture weights with are also deducted [45].
2.4. DECODING

Maximum likelihood linear regression

Techniques that only update distributions for which observations occur in the adaptation data, such as those using MAP estimation described above require relatively large amount of adaptation data to be effective.

The other approach is to estimate a set of transformations that can be applied to the model parameters. If these transformations can capture general relationships between a feature space of the original model set and one of the current speaker or new acoustic environment, they can be effective in adapting all the HMM distributions. One such transformation approach is maximum likelihood linear regression (MLLR) which estimates a set of linear transformations for the mean parameter of a Gaussian mixture HMM system to maximize the likelihood of the adaptation data [46, 47]. If only the means are considered then, an affine transform \((A_s, b_s)\) is applied to the Gaussian mean vector to map from speaker or environment-independent \(\mu_{jm}\) to speaker or environment-dependent \(\hat{\mu}_{jm}\) means:

\[
\hat{\mu}_{jm} = A_s \mu_{jm} + b_s
\]

(2.24)

where \(A_s\) is a transformation matrix and \(b_s\) is a bias vector. Transformation parameters \((A_s, b_s)\) are estimated so that they maximize the likelihood of given adaptation data.

2.4 Decoding

2.4.1 Viterbi decoding

As shown in Equation(2.2), the goal of the ASR system is to find the most probable word sequence of a given observation sequence. The state sequence, which represents the word sequence, can be obtained efficiently by the Viterbi algorithm [48]. The Viterbi algorithm can be described as follows using a new variable \(\alpha_{j}^{Viterbi}(n)\) called a Viterbi score.

\[
\alpha_{j}^{Viterbi}(n) = \max_{r \in R_{n-1}} \mathcal{L}(q(n) = j, O_{1:n}, r)
\]

(2.25)
where $R_{n-1}$ is the set of all partial paths of length $n - 1$ and $\mathcal{L}(\cdot)$ denotes the likelihood. Values of $\alpha^\text{Viterbi}_j(n + 1)$ can be efficiently calculated using the following recursive equation

$$
\alpha^\text{Viterbi}_j(n + 1) = \max_{1 \leq i \leq N_q} \alpha^\text{Viterbi}_i(n) a_{ij} b_j(o(n + 1)).
$$

2.4.2 Beam search

In LVCSR consisted with a large number of states $N_q$, it is not practical to propagate all nodes at every frame. In order to reduce the computational load, the path is pruned from the list of possible path if a Viterbi score of a path falls below a certain threshold. The threshold is called beam width and the search is called beam search. Typical beam search sets the threshold at some fixed beam width below the current most likely path. This pruning reduces the computational cost dramatically though it may introduce search errors.

2.4.3 Language multiplier

When one puts the acoustic models and the language models together, it is noted that the difference in the probability obtained from these models. The emission probabilities $b_j(o(n))$ obtained from a continuous density model such as a Gaussian mixture model are calculated as probability densities. Dynamic ranges of probabilities obtained from these density functions are noticeably varied depending on their variances. On the other hand, transition probabilities $a_{ij}$ and N-gram probabilities $P(w_m|W_{m-1:m-N_{\text{gram}}+1})$ are the probabilities normalized so that the sum of these probabilities is 1.0. Moreover, the difference in the accumulation unit must be taken into account. The unit of acoustic score is a frame, and beside, the unit of language score is a word. Typical LVCSR system introduces several multipliers or scaling factors to balance these differences. The grammar multiplier is commonly incorporated in the scores obtained from the language model. Thus, the likelihood of word $w_m$ following a history $W_{m-1:m-N_{\text{gram}}+1}$ associated observed data $O$ is
given by
\[
\log \mathcal{L}(O, w_m | \mathbf{W}_{m-1:m-N_{gram+1}}) = W_{ac} \log \mathcal{L}(O | w_m) + W_{lang} \log P(w_j | \mathbf{W}_{m-1:m-N_{gram+1}})
\] (2.27)
where \(W_{ac}\) and \(W_{lang}\) are multipliers for an acoustic score and a language score respectively. Taking into account of reliability of the scores obtained from these models, typical LVCSR system optimizes these multipliers so as to weight heavily to a confidence model.

### 2.5 Dynamic approaches in this thesis

Figure 2.5 shows dynamic approaches proposed in this thesis. Acoustical issues in model mismatch caused by particular speakers, issues in decoding with constant language multiplier, and issues in emission probability based on a joint probability of features are addressed.

The **dynamic model selection** described in Chapter 3 reduces recognition errors caused by the mismatched acoustic models. Appropriate acoustic model is selected in real-time from among the models adapted to the acoustical environments.

The **dynamic score compensation** described in Chapter 4 dynamically change the balance between the acoustic score \(\mathcal{L}(O | w_m)\) and language score \(\log P(w_j | \mathbf{W}_{m-1:m-N_{gram+1}})\) by compensating the acoustic score in noisy environment.

The **dynamic stream weighting** described in Chapter 5 gives dynamic weights to the likelihoods of multiple feature-streams so as to increase discriminability of recognition hypotheses.
Figure 2.5: Overview of dynamic approaches proposed in this thesis.
Chapter 3

Dynamic approach selecting an acoustic model

3.1 Issues on HMM adaptation

As described in Section 2.3.3, it degrades the recognition performances, often dramatically, that the model mismatches caused by the differences between training and practical acoustic environments. Broadcast news programs consisting of various speakers and acoustical environment often cause such degradations. The automatic captioning, therefore, was restricted to sections of speech read by anchor-persons as described in Section 1.1. The reason is that the use of common gender-dependent acoustic models reduce the word accuracy to unacceptable degree if utterances of the other speakers are recognized. Computational costs are also raised because reduced discriminative ability of the acoustic model increase a number of hypothesis in the search beam. Improvement in the acoustic model handling a wider variety of speakers is demanded to extend the automatic captioning for the entire part of the programs.

An acoustic model consisting of a larger number of model parameters may cover such particular speakers or environment. It, however, requires more training data than available in the database and more computational costs than the one processed in real-time.

The model adaptations described in Section 2.3.3, therefore, are one of the efficient technique to improve the performance of such acoustical mismatched conditions. In order
to adapt such acoustic models to a wider variety of speakers, multiple sets of HMMs each adapted to a speaker or a cluster of speakers can be used. If speaker labels are available for the adaptation data, a speaker dependent HMMs can be trained easily. The clusters of speakers may be efficient to reduce complexity of the recognition system. Such cluster-dependent acoustic models could be easily created by adopting speaker clustering method if a large-scale database used in the training provided speaker labels [49, 50]. These methods generate speaker clusters to obtain adaptation data for the cluster-dependent acoustic models. By grouping utterances of several speakers, the cluster dependent models utilize relative large-scale adaptation data compared with speaker dependent models. Furthermore, they can be applied for real-time applications if speakers are specified in the test utterances.

However, if one attempts to generate speaker clusters from a large-scale database, unwanted cluster dependent HMMs for the test utterances are often created by mismatched speakers in the training database and the test utterances. On the other hand, all the test data can be clustered in advance if off-line decoding is allowed [51]. Many proposals in DARPA [8] (Section 1.1) allows such situation because their studies were focused on information retrieval. Such methods can generate appropriate number of clusters for test utterances.

Some on-line adaptation methods clusters their recognition results by detecting turn taking on the fly [52, 53] and incrementally adapt acoustic models to the following test utterances. An amount of adaptation data is, however, relative small compared with the data obtained by the clustering a large-scale database in advance, especially in the beginning of comments of a speaker. These methods are not suitable for the real-time captioning because the detection of the turn taking increases computational costs and recognition results are obtained after the detection.

Following issues still remained to be addressed in our recognition system if one improves the performance of real-time broadcast news transcription by using these technique.

- The system can not specify speakers in advance.
- The system can not specify when speaker turns in advance.
- Speaker labels are not available in a large-scale database.
3.2 Overview of the proposed dynamic selection of acoustic models

The proposed dynamic selection of an acoustic model described in this chapter is outlined in Figure 3.1. The method utilizes a short fragment of the data at the beginning of an input utterance to determine the most appropriate model from among cluster-dependent acoustic models. The selected model is the one corresponding to the cluster whose cluster model gives highest likelihood for the short fragment. The clusters are modeled by using the Gaussian mixture model (GMM) [54] (Section 2.3.1) whose likelihood calculation have a low computational cost.

This chapter describes an efficient training method of the GMMs and the cluster-dependent acoustic models for the dynamic selection. Section 3.3 presents an efficient method of clustering a large-scale speech database. Utilizing a small database representing a characteristics of a target news program, this method clusters a wide variety of adaptation utterances in two stages. Section 3.4 describes a real-time selection of an appropriate cluster-dependent...
acoustic model in its decoding stage. Section 3.5 describes experiments of Japanese broadcast news transcription and discusses the results. Further analysis of applicability to real-time news transcription system are discussed in Section 3.6.

3.3 Tow-stage clustering

We adopt a two-stage clustering to obtain adaptation utterances for each cluster dependent acoustic models efficiently. In this clustering, cluster models, which are represented by using GMMs, are employed to represent utterances in each cluster. Figure 3.2 illustrates an example of the timescales of the data used in this two-stage clustering. Test data (a) is the target news program to be recognized. The first stage clusters a small database (b) gathered from past editions of the same program as (a) and generates the program specific cluster models of GMMs. This database is carefully selected so as to represents the characteristic of a target task. The second stage selects adaptation utterances for each cluster dependent acoustic model from a large-scale database (c) of any TV programs by using the cluster models (GMMs). A speaker or environment independent acoustic model trained with large-scale database (c) are adapted for each cluster by using the selected adaptation utterances.
3.3. TOW-STAGE CLUSTERING

3.3.1 First stage of clustering

Figure 3.3 illustrate an outline of the first stage of the clustering. The small database (c) used in this stage consists of the same TV program as the target program, compiled over a period of several days. Speech data in the programs are chopped into segments and they are acoustically analyzed into feature vectors as training data on a segment basis. The $k$-th segment of the observation sequence denoted by $O^{seg}_k = \{o_k(n) : n = 1 : N^{seg}_k\}$ $(1 \leq k \leq N^{sdb})$ is assigned into cluster $c$ $(1 \leq c \leq N^C)$ by the function $C(O^{seg}_k)$, where $o_k(n)$ is a observation vector, $N^C$ is a number of clusters, $N^{seg}_k$ is a length of the observation vector sequence of segment $k$, $N^{sdb}$ is a number of segments in the small database and $C(O^{seg}_k)$ is a function assigning a segment $O^{seg}_k$ to a cluster $c$. The assignment function $C(O^{seg}_k)$ is defined by using the GMMs as follws,

$$C(O^{seg}_k) = \arg \max_c \prod_n^{N^{seg}_k} \sum_i^m c_{ci}N(o_k(n)|\mu_{ci}^{gmm}, \Sigma_{ci}^{gmm})$$, \hspace{1cm} (3.1)
where $c_{gi}^{gmm}$, $\mu_{ci}^{gmm}$ and $\Sigma_{ci}^{gmm}$ are $i$-th component of GMM parameters $\theta_c^{gmm}$ corresponding to $c$-th cluster.

The GMM parameters $\theta_c^{gmm}$ is estimated by using the EM algorithm as follows,

$$
\hat{\theta}_c^{gmm} = \arg \max_{\theta_c^{gmm}} \prod_{k=1}^{N_{seg}} \sum_{i=1}^{m} c_{ci} \mathcal{N}(o_k(n)|\mu_{ci}^{gmm}, \Sigma_{ci}^{gmm})
$$

(3.2)

This clustering procedure is followed by the repetition of two operations: aligning each segment to the most probable cluster and estimating parameters of GMMs. The detailed procedure is as follows.

1. Assign each segment to any cluster randomly. The number of mixtures in the initial GMM is set to 1.

2. Estimate parameters $\theta_c^{gmm}$ of the GMMs with the segment assigned to the cluster by using the EM algorithm and Equation (3.2). The estimation is iterated $Ite_{em}$ times.

3. Reassign each segment to the cluster which gives the highest GMM likelihood among all the clusters by using Equation (3.1).

4. Repeat the procedure (2)-(3) until no migration of segments among the clusters occurs or a specified number of iteration $Ite_{align}$ is reached.

5. Increment the number of mixtures and repeat the procedure (2)-(4). When the number of mixtures reaches the desired figure $N_{mx}$, terminate the first stage of clustering.

### 3.3.2 Second stage of clustering

The second stage uses the large-scale database (c) compiled from many types of programs including the target program to cover a wide variety of speakers. The outline of the second stage of the clustering is illustrated in Figure 3.4. The utterances in the large-scale database are also analyzed into feature vectors on a segment basis. Each one is merged into the cluster which gives the highest GMM likelihood among all the clusters if the likelihood exceeds a threshold. The threshold is set to $\nu_c - 2\rho_c$ where $\nu_c$ is the segment-averaged
3.3. TOW-STAGE CLUSTERING

Figure 3.4: Outline of the second stage of the clustering.

likelihood and $\rho_c$ is the standard deviation of cluster $c$. It corresponds to a 97% interval for each cluster. Segments which are unnecessary for adapting cluster-dependent HMMs are discarded due to this threshold. The second stage of clustering is performed faster than conventional direct clustering methods because it requires only the likelihood computation of each segment against each GMM.

3.3.3 HMM adaptation

After selecting segments for each cluster, parameters of gender-dependent HMMs are adapted to each cluster by the MAP [44, 45] estimation. The reasons why the MAP adaptation is adopted are described as follows,

- There may not be enough data to train HMMs in a maximum likelihood way when the number of cluster is larger.

- The same tying structure is advantageous when switching to the most appropriate cluster-dependent HMM in decoding because it requires only the change of Gaussian parameters (means, variances and weights) without changing a tree-structured sub-word network of tied HMMs (Section 2.3.2).

The HMMs re-estimated here can be called cluster dependent HMMs.
3.4 HMM selection using GMMs

The broadcast news captioning system is required to output a result almost instantaneously. The cluster dependent HMMs is also expected to improve a real-time factor (RTF) because improved discriminability provided by the cluster dependent HMMs prune more hypotheses under the same search condition such as the same search parameters of beam width. The time to start a decoding, therefore, can be delayed for some extent so that the most appropriate model is selected. This method utilizes a short fragment of speech data at beginning of an input utterances for the determination of appropriate cluster dependent acoustic model. A length of a speech fragment $\Delta t$ must be long enough to maintain the accuracy of the model selection.

A selected set of HMMs is the one whose cluster model (GMM) gives the highest likelihood for a short fragment of an input utterance. The model selection based on the GMMs has a low computational cost because it just compares several likelihoods of the GMMs but it delays starting the decoding by the length of the fragment.

3.5 Experiments

3.5.1 Setup

We carried out Japanese broadcast news transcription experiments. Table 3.1 summalizes sizes of speech data used in this experiments. For the evaluated speech data ((a) in Figure 3.2), we used 335 utterances consisting of 8,489 words in total. They were chosen from four news programs aired on September 30th 1998 and uttered by 7 anchormen and 7 male reporters.

The first stage of clustering used 3,091 utterances in the same four news programs aired from September 20th through 29th in 1998 ((b) in Figure 3.2). The GMMs, whose number of Gaussian components was 32, were trained from these utterances. In this experiments iterative parameters of $Ite_{em}$ was set to 3 and $Ite_{mx}$ was set to 3. This experiment compared 5 sets of the GMMs whose number of clusters were 2, 4, 8, 12, or 16.

As a large-scale database for the second stage of clustering ((c) in Figure 3.2), we used
68,101 utterances in broadcast news programs aired from June 1996 through September 29th 1998.

All the speech segments used in these experiments were analyzed into 39 parameters consisting of 12 MFCCs with log-power and their first- and second-order regression coefficients. These features were calculated every 10msec after digitizing at 16kHz and 16bits with a Hamming window of 25msec width.

The baseline acoustic model was trained from the large-scale database (c). It consisted of state-tied triphone HMMs trained by tree-based clustering with 42 Japanese phonemes. The numbers of HMMs were 5,898 logical and 3,441 physical consisting of 2,787 states for a lexicon of 20k words. This model were adapted to the cluster dependent models by using the MAP estimation. The weighting factor $\tau$, which determines a degree of the affection by the adaptation data as described in Equation 2.23, was set to 10. The mixture weights, means, and variances of the HMM parameters were updated by this adaptation.

The decoder used in this experiment was a following 2-pass decoder [55]. The first pass of the decoder generates a word lattice time-synchronously by Viterbi beam search using bigram language model and triphone HMMs. At a sentence-end, the word lattice is recursively traced back to get N-best sentences. The second pass re-scores the N-best sentences by a trigram language model to decide and output the best sentence.

The language models used in this experiment was the time dependent language model (TDLM) [56]. This model was trained from NHK’s news scripts extending back over 7 years with latest news more heavily weighted after Japanese morphological analysis. Four different TDLMs were trained for corresponding four news programs in the test set. The
perplexities of the trigram models for the evaluated utterances were 11 to 54. The rate of words outside of the vocabulary was 0.75% to 1.13%. Real-time factors are measured by using an Alpha-21264 500MHz machine with 1G byte memory.

Speakers included in each database

The small database generating the cluster models of the GMMs were uttered by 42 speakers consisting of 15 anchormen and 27 reporters. The speakers common among the test set and the database consist of 6 anchormen and 3 reporters. Therefore, segments uttered by 5 speakers consisting of an anchorman and 4 reporters were not used for the training of the GMMs.

There were 18 speakers who talk in the news programs regularly. They were 15 anchormen and 3 reporters and their utterances were included in the small database. The test set included 6 regular anchormen and 2 regular reporters. It was supposed that there were many segments uttered by the regular speakers but were not many segments uttered by the others.

3.5.2 Baseline recognition result

The recognition results were evaluated by the word error rates (WER) and real-time factors (RTF). The WER is the percentage of error words consisting of substitution, insertion, and deletion errors to the total words of the reference.

\[
WER = \frac{S + I + D}{N}
\]

(3.3)

where \(N\) is the total number of the reference words, \(S\) is the number of words substituted in the recognition results, \(I\) is the number of words inserted in the recognition results, and \(D\) is the number of words deleted in the recognition results. The RTF is the ratio of the time required for the decoding to the length of the input segments.

\[
RTF = \frac{T_{Proc}}{T_{In}}
\]

(3.4)
where, $T^{Proc}$ is the time required for the decoding and $T^{In}$ is the length of the input segments. Following experiments were compared with the baseline acoustic model whose WER was 7.95% and the RTF was 0.96.

### 3.5.3 A number of clusters

At first, the optimum number of the clusters were examined. By changing the number of clusters from 2 to 16, five sets of cluster dependent acoustic models were compared. In this experiment, the best acoustic model was selected by using the likelihood calculated from whole utterance. In this experimental setup, the time to start a decoding was delayed for a length of an input utterance because the model selection was done after observing the fragment $\Delta t$. The recognition, therefore, did not work in real time. However, it is probably the least likely case the model selection would make an error.
The result of this experiment is illustrated in Figure 3.5. White dots denote WER on the left axis and back dots denote RTF on the right axis. The result showed that WER was reduced as the number of the clusters increased from 2 to 12 but the 16-cluster system showed worse WER than the 12-cluster system. We could conclude that the best number of cluster among this test case was 12. In the best case, a 20% reduction of WER compared with the baseline method was obtained. The result for RTF showed the same tendency as the result for WER. The most effective number of clusters for RTF was also 12 with a 23% reduction. The reason why the cluster dependent acoustic models which reduced WER also improved RTF was the search algorithm adopting beam pruning. By using a better acoustic model, the number of confusable words within the likelihood threshold was reduced.

3.5.4 Length of a fragment to determine an acoustic model

In this experiment, we tried to decrease the length of the fragment for the model selection in order to reduce the total time for the model selection and decoding so as to obtain real-time operation. The fragment length $\Delta t$ at the beginning of input segment was varied from 0.1 to 2.0 seconds. The decoding started after the selection of the cluster dependent acoustic models (HMMs) corresponding to the most probable cluster model (GMM) according to the $\Delta t$ fragment. In this experiment, a RTF was evaluated by the processing time $T_{\text{Proc}}$ calculated as the sum of the length of the fragment $\Delta t$ and the time required for the decoding.

WERs and RTFs obtained in this experiment are illustrated in Figure 3.6. The fragment length $\Delta t$ used in the model selection is displayed in the abscissa axis. White dots denote WER on the left axis. Black dots denote RTF on the right axis with taking into account of the time required to get a fragment $\Delta t$. The RTF linearly increased as the $\Delta t$ increased. On the other hand, The WER did not change much when the fragment length was between 0.5 seconds and 2.0 seconds compared with the case when selection was done on the whole segments. A fragment length of less than 0.5 seconds, however, degraded the WER due to miss-selection of acoustic models. Therefore, we conclude that the best fragment length was 0.5 seconds which gave 20% reduction of WER and 22% reduction of RTF compared with the baseline method.
3.5. EXPERIMENTS

Figure 3.6: Word error rates and real-time factors in each fragment length when the number of clusters was 12.

Figure 3.7 illustrates WER on a varied fragment length for different numbers of clusters from 2 to 16. The result showed that the fragment length required for a stable model selection was decreased as the number of clusters decreased. The system consisting of 12 clusters, however, showed the best performance at a fragment length of less than 0.5 seconds. It is concluded that 12 cluster system was the best system even if the fragment length had to be set to less than 0.5 seconds.
3.6 Further analysis

3.6.1 Efficiency of the two-stage clustering

Computational cost

This section gives further analysis of the proposed method. At first, the computational cost required for the two-stage clustering are discussed. The computational cost is evaluated by the number of likelihood calculations of Gaussian components given a number of segments $N_{ldb}$ in the large-scale database, a number of Gaussian component $N_{mx}$ of the GMMs, and a number of clusters $N^C$. If $N_{mx} \gg 1$, the computational cost of the first stage is proportional to

$$C^{1st stage} = \frac{1}{2}((N_{mx})^2 + N_{mx} - 2)It_{cm}It_{mx}N^CN_{ldb},$$

(3.5)
3.6. FURTHER ANALYSIS

Table 3.2: Comparisons of the proposed clustering with conventional clusterings.

<table>
<thead>
<tr>
<th>Clustering</th>
<th>WER [%]</th>
<th>RTF</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baseline</td>
<td>9.43</td>
<td>1.03</td>
</tr>
<tr>
<td>Conventional</td>
<td>6.71</td>
<td>0.86</td>
</tr>
<tr>
<td>1st stage</td>
<td>7.04</td>
<td>0.79</td>
</tr>
<tr>
<td>Threshold</td>
<td>6.78</td>
<td>0.79</td>
</tr>
<tr>
<td>Exclusive</td>
<td>6.38</td>
<td>0.77</td>
</tr>
<tr>
<td>Proposed</td>
<td>6.37</td>
<td>0.74</td>
</tr>
</tbody>
</table>

and the computational cost of the second stage is proportional to

\[ C^{2nd stage} = N^{mx} N^C N^{ldb}. \]  \hspace{1cm} (3.6)

If the large-scale database was directly clustered using the method of the fist stage, the computational cost would be proportional to \( C^{1st stage} \). The computational cost of the proposed two-stage clustering is

\[ C^{proposed} = \frac{N^{sdb}}{N^{ldb}} + \frac{1}{(N^{mx} + 1 - 2/N^{mx})Ite_{mx}Ite_{em}} \]  \hspace{1cm} (3.7)

times as much as the cost required the direct method.

In this experimental condition, where \( N^{mx} = 32 \), \( Ite_{mx} = 3 \), \( Ite_{em} = 3 \) and \( \frac{N^{sdb}}{N^{ldb}} \simeq 0.1 \), the second term of the ratio is negligible (\( \simeq 0.003 \)); therefore the two-stage clustering reduced 90% of likelihood calculations compared to the direct clustering.

WER comparison with conventional clustering

Secondly, WER of the proposed clustering were compared with conventional methods of clustering. Following clustering methods are compared in this section.

Conventional This method directly clusters the large-scale database using the clustering method of the first stage.
**1st stage**  This method obtains the adaptation utterances from the 1st stage of the clustering. Thus, the large-scale database was not used.

**Threshold**  This method allows segments being selected for multiple clusters in the 2nd stage of the clustering. In other words, each segment in the large-scale database is assigned into clusters if their likelihoods exceeds the threshold.

**Exclusive**  This method does not use the threshold in the 2nd stage of the clustering. Therefore, each segment in the large-scale database is always assigned into a cluster.

Table 3.2 compares WER and RTF of the gender dependent acoustic model (Baseline) and the propose model (Proposed) with the models described above. Number of segments assigned to each cluster at first- and second- stage in the 12 cluster system is compared in Table 3.3. The results showed that the result of Conventional was better than the result of 1st stage because Conventional obtained more segments than 1st stage for the adaptation. However, Proposed showed better result than Conventional though the difference in amounts of adaptation segments were small. It is supposed that the difference comes from Proposed utilizing small database representing the characteristic of target programs. It is concluded that the proposed method improved WER and RTF by utilizing program specific database.

The results also shows that the result of Exclusive was better than the result of Threshold. Moreover, Proposed reduced 4% of RTF compared to Exclusive though there is no meaningful difference in WER. It is, therefore, concluded that the another dominant factor in the improvement of WER obtained from the proposed method was exclusiveness of the 2nd stage of the clustering. The threshold in the 2nd stage improved RTF by discarding unwanted segments for the evaluation.

**3.6.2 Speakers in the clusters**

This section discusses the relationship between the GMMs and speakers. Distributions of segments uttered each speaker to the clusters are shown in Figure 3.8. The figure shows a number of assigned segments using density of hatching. It shows 14 speakers consisting of 7 anchormen and 7 reporters in the horizontal axis and assigned clusters in the vertical
3.6. FURTHER ANALYSIS

Figure 3.8: Distribution of segments uttered each speaker to the clusters.

Table 3.3: Number of adaptation segments.

<table>
<thead>
<tr>
<th>Cluster ID</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
</tr>
</thead>
<tbody>
<tr>
<td>First stage</td>
<td>142</td>
<td>341</td>
<td>276</td>
<td>187</td>
<td>254</td>
<td>158</td>
</tr>
<tr>
<td>Second stage</td>
<td>428</td>
<td>7745</td>
<td>5736</td>
<td>5517</td>
<td>11985</td>
<td>4387</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Cluster ID</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>11</th>
<th>12</th>
</tr>
</thead>
<tbody>
<tr>
<td>First stage</td>
<td>410</td>
<td>302</td>
<td>51</td>
<td>102</td>
<td>243</td>
<td>292</td>
</tr>
<tr>
<td>Second stage</td>
<td>7416</td>
<td>4172</td>
<td>1431</td>
<td>4096</td>
<td>10993</td>
<td>3270</td>
</tr>
</tbody>
</table>
Table 3.4: Comparison of the recognition results of regular speakers and the others.

<table>
<thead>
<tr>
<th>Speaker</th>
<th>Baseline</th>
<th>Proposed</th>
<th>Reduction</th>
</tr>
</thead>
<tbody>
<tr>
<td>1,2,3,8,9</td>
<td>6.3%(0.76)</td>
<td>4.8%(0.53)</td>
<td>23% (30%)</td>
</tr>
<tr>
<td>Others</td>
<td>11.4%(1.06)</td>
<td>10.9%(0.95)</td>
<td>11% (10%)</td>
</tr>
<tr>
<td>Open</td>
<td>11.4%(1.20)</td>
<td>10.4%(1.09)</td>
<td>9% (9%)</td>
</tr>
</tbody>
</table>

axis. Speaker IDs from 1 to 7 shows segments uttered by 7 anchormen and IDs from 8 to 14 shows segments uttered by 7 reporters. The cluster dependent models whose IDs are 11 and 12 were not used for the decoding of the test set. They are supposed to be the clusters of the speakers who uttered in the small database but did not utter in the test set. The speakers whose ID are 1, 2, and 3 were main anchormen of the news program and the speakers whose ID are 8 and 9 were weather reporters. They were regular speaker of the program. It is supposed that speaker dependent acoustic models were constructed for these regular speaker.

Table 3.4 shows WER, RTF and their reduction rates of the regular speakers (ID=1, 2, 3, 8, 9) and the other speakers. The results showed that the proposed method improved both of these speakers, especially in the regular speakers. The test set included 5 speakers who were not included in the small database of the 1st stage of the clustering. Their speaker IDs were 5, 10, 11, 13, and 14. The result of these speakers is shown in the row “Open”. The proposed method also improved the result of these speakers similarly to the result of “Others”. It is concluded that the proposed method was not only effective for the speakers in the small database but also the open speakers.

### 3.6.3 Improvement of each cluster

Figure 3.9 compares WER and the reduction rate relative to the baseline of each cluster dependent model. The white bars denote WER of baseline acoustic model and the black bars denote WER of the proposed cluster dependent models on the right axis. The white dots denote error reductions obtained by the proposed model on the left axis. The results showed that the proposed model did not always improve the performance because the results were
3.6. FURTHER ANALYSIS

Figure 3.9: Error reduction rate in each cluster.

Figure 3.10: Shows the frame averaged log-likelihood of each cluster model (GMM). They are calculated from the test utterances. The lowest likelihood was obtained from the cluster 5 from among the clusters form 1 to 5, which were mainly assigned the segments uttered by anchormen. Similarly the lowest likelihood was obtained from the cluster 10 from among the clusters from 6 to 10, which were mainly assigned the segments uttered by reporters. It is supposed that the cluster dependent model whose GMM gives lower likelihood may degrade the performance. Therefore, a method addressing these degradation will be required.

3.6.4 A comparison between anchors and reporters

Figure 3.11 shows reduction rates of WER and RTF of two speaker categories consisting of anchormen and reporters. A larger reduction rate was obtained from Reporter in both of WER and RTF. The baseline acoustic model was well matched to the utterances by
CHAPTER 3. DYNAMIC APPROACH SELECTING AN ACOUSTIC MODEL

Figure 3.10: Averaged GMM likelihood of each cluster.

Figure 3.11: WER and RTF reduction by the speakers.
the anchornen because the stochastic model was trained from the large-scale database whose ratio of the segment uttered by the anchornen is obviously larger than the segment uttered by the reporters. On the other hands, the reporters consisting of more speakers than the anchornen need adaptation to improve the recognition performance. Therefore, the proposed method is supposed to reduce these mismatches and improved the recognition performance of the reporter’s speech.

3.7 Concluding remarks

This chapter presented a dynamic approach of selecting an appropriate acoustic model in real time. The method uses a short fragment of the data at the beginning of the input utterances to determine the most appropriate model from among cluster-dependent acoustic models, which are adapted for clusters of acoustical environments such as speakers or noisy conditions. The model selection utilizes GMMs corresponding to the clusters and selects an acoustic model whose GMM gives the highest likelihood for a short fragment.

The efficient training methods of the GMMs and the cluster dependent models were also addressed in this chapter. The method clusters a large-scale database in two stages for the trainings of GMMs and acoustic models. The first stage obtains GMMs from a small database representing characteristics of target programs. The second stage obtains adaptation data for the cluster-dependent model form a large-scale database. The proposed clustering algorithm reduces computational cost compared to the conventional single-stage clustering.

A transcription experiment implementing the proposed method for Japanese broadcast news for the purpose of automatic captioning showed that the method obtained the maximum improvements in word error rate and real-time factor when a number of clusters was 12. The best system determined the appropriate acoustic model with a 0.5 seconds fragment of the input utterance and reduced 20% of error word and 22% of real-time factor relative to the baseline system.

Further analysis also showed that the proposed method improved WER and RTF by utilizing program specified database in its 1st stage, the exclusiveness of its 2nd stage improved WER, and the threshold of its 2nd stage improved RTF. The method improved
the recognition results of the regular speakers by constructing speaker-dependent acoustic models and the method also improved the results of open speakers. It is remarkable that the method improved the recognition performance of the reporter’s speech which are strongly demanded to reduce the mismatch between the speech and the models.

Future work will involve experiments introducing the on-line adaptation technique for new speakers to develop more robust speech recognition.
Chapter 4

Dynamic compensation of acoustic scores

4.1 Issues on reliability of acoustic scores

The primary issue addressed in the previous chapter was the mismatch caused by the difference in speakers between training speech database and input utterances. In this chapter, mismatches caused by unexpected background noise are addressed. These kinds of mismatches often degrade the performance of the speech recognition, especially in sports programs. The sports programs, which have relatively high audience ratings, place a significant demand for segment metadata to be included in live broadcasting contents. The extraction of the metadata describing “who did what and when” is strongly expected to be assisted by the speech recognition technology in such live broadcasting as described in Section 1.2.

However, the complexity of the background noise of such live sports broadcasting makes it difficult to recognize them with a sufficient degree of accuracy. A limited amount of adaptation data makes it difficult to train a wide variety of background noises. A limitation on the model complexity allowed for the real-time processing, that is to say, a limited number of continuous density components of the HMM states, also makes it difficult to cover such a variety of background noises. In fact, statistics of observed features for every phoneme can not be yielded in all noisy environments. Especially, the statistics of the
phonemes in minor noise or burst noise such as applauds by the audience are not well modeled in a statistical acoustic model. Thus, conventional HMM adaptation techniques such as MLLR [46, 47] can not improve recognition accuracy enough for the closed captioning or the metadata extraction. Furthermore, a rapid change in background noise make it difficult to estimate the noise’s characteristics. The noise estimation error caused by such unstable background noise hinders feature space compensation techniques such as spectral subtraction [35, 36] from improving the accuracies of the recognition.

On the other hand, weighted Viterbi decoding, which changes the weight of an acoustic score calculated frame by frame by using a confidence factor, improves digit recognition of Aurora2 database [57, 58] and broadcast news transcription [59] in noisy environments. They compensates acoustic scores by multiplying the confidence factor to each HMM likelihood dynamically, that is to say, the method use the confidence factor as the acoustic multiplier in Figure 2.5. The confidence factor is calculated so that the method gives lower acoustic weight to an unreliable input frame which may degrade the performance of the recognition. This chapter addresses the unreliability of the input frames in noisy environments.

The method proposed by Cui et al. [58] used the confidence factor based on a estimation of SNR calculated by the minimum statistics tracking method [60]. The method estimates a frame-based background noise spectrum and an SNR by tracking power spectral minimum within a 0.5 second interval preceding each speech frame. This algorithm can deal with non-stationary background noise whose characteristics change statistically slowly without any training data of the noise. However, it has difficulty of estimating a frame-based SNR when rapidly changing background noises exist and the estimation errors may degrade a performance of the speech recognition. The practical applications are, therefore, restricted because of the estimation errors coursed by a mismatch between the assumption of the spectral power of the noise changing slowly and the practical input noises.
Figure 4.1: Overview of the proposed dynamic compensation of acoustic scores
4.2 Overview of the proposed dynamic compensation of acoustic scores

The proposed method calculates a frame-wise confidence factor from the noise models incorporating prior knowledge of various noises. The proposed confidence factor, in contrast to the factor based on the SNR estimated by a rather simple method [58], is calculated by using the noise models of spectral shapes and their dynamics, that is to say, MFCCs and their regressions. Utilizing the noise models trained for each noise category, the method captures the unreliability caused by noise even if it changing rapidly.

Figure 4.1 outlines the proposed dynamic compensation of acoustic score. The likelihood of the noise models plays two roles in this method. Firstly, it gives a confidence factor of an input frame so as to weight and compensate an acoustic score. It is expected that the compensated acoustic score will improve robustness to background noise of an input speech because a lower confidence factor compresses acoustic scores for a noisy frame, in which equivalently the decoder rather relies on a language score and keeps more hypotheses within a fixed search depth.

Secondly, the likelihood of a noise model is used as an alternative that of a silence HMM. The noise models are not trained with any speech to be recognized; therefore these models can be regarded as alternative models of the silence HMM. This alternation of the likelihood is expected to give more reliable acoustic scores to non-speech parts than those obtained by the single silence HMM. Further information about the non-speech parts is provided by the models of noise categories, in contrast to the information provided by a general noise model of a silence HMM, is expected to improve the recognition performance.

In addition to these techniques, the method can be used together with the conventional noise withstanding techniques of subtraction and adaptation.

Section 4.3 of this chapter describes the proposed method of compensation. Section 4.4 reports the results of baseball broadcasting transcription experiments. Section 4.5 gives further investigations and Section 4.6 concludes this chapter by briefly summarizing it.
4.3 Acoustic score compensation

As shown in Figure 4.1, this method uses speech models and noise models of Gaussian mixture models (GMM) trained beforehand with appropriate speech and noise data, respectively. Their likelihoods are calculated for each input frame and used in two ways in this method.

1. The confidence factor of each input frame is obtained by using the noise likelihood and the speech likelihood. Then the acoustic score of each HMM state in the Viterbi path is compensated with the confidence factor.

2. A likelihood of the noise model is alternatively used for a likelihood of the silence HMM if the likelihood of the noise model exceeds that of the silence HMM.

The detailed procedure is described in the following sections.

4.3.1 Noise and speech models

The proposed method uses two sets of GMMs trained in advance. One is a set of noise models trained with data consisting of many kinds of background noise expected to appear in the broadcast sports games to be recognized, such as a crowd noise in a stadium. The training noise data is collected from the past broadcasting sports games so that the data consists of miscellaneous noise features.

In the collected noise data, the occurrence frequencies of the noise categories are different from each other and the differences are often represented by the model parameters of mixture weights of Gaussian components representing each noise category if a single noise model is trained with the all noise data. A Gaussian component of infrequent minor noise may have a smaller mixture weight and a larger variance, so that the component barely contributes to noise likelihood. The multiple noise models are, therefore, trained separately for every noise category or cluster. By reducing the dependence on the occurrence frequency of the noise, the multiple noise models yield higher likelihood than that yielded by the single noise model even if an input noise is the minor noise. They also increase discriminative ability of the noise against the speech.
The set of noise models \( \lambda^n = \{ \lambda^n_k; 1 \leq k \leq N^n \} \), where \( \lambda^n_k \) is a noise model corresponding to \( k \)-th noise cluster and \( N^n \) is a number of the noise clusters, are GMMs trained with the data in the noise clusters. These noise models are used to obtain the confidence factors and the alternative likelihood of the silence HMM.

The other set of GMMs comprises the speech models for the confidence factor calculation. They are small models approximating the large acoustic model of triphone HMMs so as to calculate the confidence factor in real-time. The GMMs are trained from the speech database from which the acoustic model (HMMs) is trained. For an easier implementation, they are trained on phonemes. Thus, the set of speech models \( \lambda^s = \{ \lambda^s_l; 1 \leq l \leq N^s \} \) where \( \lambda^s_l \) is a speech model corresponding to \( l \)-th phoneme and \( N^s \) is a number of the speech clusters, are GMMs of phonemes.

The likelihood \( P(o(n)|\lambda^n_k) \) of each noise model and the likelihood \( P(o(n)|\lambda^s_l) \) of each speech model are calculated for each MFCC-based feature vector \( o(n) \) for frame \( n \). The noise likelihood \( P(o(n)|\lambda^n) \) and the speech likelihood \( P(o(n)|\lambda^s) \) are the maximum likelihoods of the corresponding GMMs as follows,

\[
P(o(n)|\lambda^n) = \max_k P(o(n)|\lambda^n_k), \quad (4.1)
\]
\[
P(o(n)|\lambda^s) = \max_l P(o(n)|\lambda^s_l). \quad (4.2)
\]

### 4.3.2 Confidence factor

The confidence factor is calculated from the noise models and the speech models frame by frame. In this method, a posterior probability of the speech models \( P(\lambda^s|o(n)) \) given input feature \( o(n) \) is adopted to the confidence factor in order to decrease the factor for a noisy frame. The confidence factor \( \mathcal{W}^{ac}(n) \) is defined as follows,

\[
\mathcal{W}^{ac}(n) \triangleq P(\lambda^s|o(n)) = \frac{P(o(n)|\lambda^s)P(\lambda^s)}{P(o(n))} = \frac{P(o(n)|\lambda^s)}{P(o(n)|\lambda^s) + P(o(n)|\lambda^n)}, \quad (4.3)
\]

assuming

\[
P(\lambda^s) = P(\lambda^n) = 0.5 \quad (4.4)
\]
and
\[ P(o(n)) = 0.5 \{ P(o(n)|\lambda^s) + P(o(n)|\lambda^n) \} \] (4.5)

### 4.3.3 Compensated acoustic scores

The recursive equation, as described in Equation (2.26) in Section 2.4.1,
\[ \alpha^\text{Viterbi}_v(n+1) = \max_{1 \leq i \leq N_q} \alpha^\text{Viterbi}_i(n) a_{ij} b_j(o(n+1)) \] (4.6)
gives the accumulated Viterbi score where \( \alpha^\text{Viterbi}_j(n) \) is a Viterbi score of state \( j \) in the Viterbi path for frame \( n \), \( a_{ij} \) is the transition probability from state \( i \) to \( j \), and \( b_j(o(n)) \) is an observation probability of state \( j \) given observation \( o(n) \).

The proposed method compensates the acoustic score \( b_j(o(n)) \) for all \( j \) for noisy input \( o(n) \) by \( W_{\text{ac}}(n) \). This compensation compresses the dynamic range of the acoustic scores for noisy frames, so that the Equation (4.6) is modified as follows,
\[ \alpha^\text{Viterbi}_v(n+1) = \max_{1 \leq i \leq N_q} \alpha^\text{Viterbi}_i(n) a_{ij} b_j(o(n+1)) W_{\text{ac}}(n) \] (4.7)

This compensation equivalently makes the grammar multiplier, which is described in Section 2.4.3, larger at a less confidence frame where \( W_{\text{ac}}(n) \rightarrow 0 \). If the decoder limits the search depth by using a fixed beam width, as described in Section 2.4.2, a smaller confidence factor \( W_{\text{ac}}(n) \rightarrow 0 \) makes the search depth deeper than that compensated by a larger confidence factor \( W_{\text{ac}}(n) \rightarrow 1 \). Namely, at a very noisy frame where \( W_{\text{ac}}(n) \simeq 0 \), the Viterbi paths are extended without pruning based on acoustic scores. Only language scores, therefore, is taken into account at such a frame. These properties of the method are expected to improve the recognition accuracy in the noisy environment by reducing contributions of the acoustic likelihoods of such noisy frame to the accumulated scores \( \alpha^\text{Viterbi}_j \).
4.3.4 Silence likelihood alternation

A silence HMM of the acoustic model is a general noise model whose variances are relatively large and mixture weights are affected by the occurrence frequencies of noise classes. Therefore, sufficient likelihood to decode a silence may not be obtained from the silence HMM, especially for infrequent minor noise classes.

On the other hand, a higher likelihood is expected to be obtained from the noise model \( \lambda_n^k \) corresponding to the class of input noise \( k \) than the general noise model because the noise models, whose variances are smaller than that of the general noise model, are trained independently from occurrences of noise classes.

The proposed method considers \( \lambda_n^k \) to be an alternative silence HMM of the acoustic model. The likelihood of the silence HMM \( b_j = \text{silence}(o(n)) \) can therefore be calculated in decoding as follows,

\[
\hat{b}_j(o(n)) = \begin{cases} 
\max \{ b_j(o(n)), P(o(n)|\lambda_n^k) \} & \text{if } j = \text{silence} \\
b_j(o(n)) & \text{otherwise}.
\end{cases}
\] (4.8)

Finally, Equation (4.7) is modified by this alternation as follows,

\[
\alpha_{Viterbi}^V(n + 1) = \max_{1 \leq i \leq N_q} \alpha_{i}^{Viterbi}(n) a_{ij}^V \hat{b}_j(o(n + 1))^{Wac(n)}.
\] (4.9)

4.4 Experiments

We carried out transcription experiments on broadcast sports commentary. In this experiments, major league baseball (MLB) games were examined for the purpose of captioning and metadata extraction.

4.4.1 Setup

We select Japanese 761 segments uttered by a male announcer during a broadcast MLB game (New York Yankees vs. Seattle Mariners aired on 10th August 2003) as an evaluation set. This set included 8,038 words and they were automatically chopped into segments with
the frame power basis.

An acoustic model was trained with 473 hours of Japanese broadcast news speech and was adapted to broadcast MLB games with MLLR [46,47] described in Section 2.3.3 and MAP [44,45] described in Section 2.3.3. The adaptation data were gathered form six broadcast MLB games other than the test set game. The adaptation games were also chopped into segments automatically by utilizing their frame power. These adaptation segments can be classified into three classes:

- only noise (1.5 hours)
- clean speech (7.7 hours)
- noisy speech (5 hours)

Following two acoustic models, $\theta^c$ and $\theta^n$, were adapted with different set of adaptation speech and compared in this experiment.

$\theta^c$ adapted with clean speech only.

$\theta^n$ adapted with both of clean speech and noisy speech.

The HMMs used in this experiment were three-state left-to-right models of Japanese triphones and the set of HMMs consists of 4k tied-states. The number of Gaussian components of all states of the HMMs and GMMs was fixed to 24.

The speech GMMs of $\lambda^s$ were the Gaussian mixture of all 123 states of the monophone HMMs (3 states of each 41 Japanese phonemes). These HMMs were trained in the same manner as the triphone HMMs except for the model size. Thus, two sets of GMMs $\lambda^s$ corresponding to $\theta^c$ and $\theta^n$ were trained.

On the other hand, the noise segments of 1.5 hours were manually divided into following seven classes:

- crowd noise (260 segments)
- announcements over public address systems (188 segments)
- background music over public address systems (246 segments)
• the 7th inning stretch (48 segments)

• sounds of batting (218 segments)

• sounds of catching (412 segments)

• others (1,670 segments)

The noise GMMs of \( \{ \lambda_k^n; 1 \leq k \leq N^n = 7 \} \) corresponding to these seven classes were trained from these sets of segments.

The feature vector \( o(n) \) is composed of 39-dimensional parameters: 12-dimensional MFCC (see Section 2.2.1) vectors with log-power and their first- and second-order regression coefficients obtained every 10msec with a Hamming window of 25msec.

The language model with a vocabulary size of 29,501 words was trained from broadcast sports news and broadcast MLB game transcriptions (217,260 segments consisting of 2,932,267 words). The perplexity of the trigram model for the evaluated segments was 82.3. The out-of-vocabulary word rate was 2.3%.

We used a two-pass decoder as follows [10]. The first path, using a bigram language model and triphone HMMs, generates a word lattice time-synchronously by using the Viterbi beam search. At the end of the sentences, the word lattice is recursively traced back to get the N-best sentence hypotheses. The second path re-scores the N-best sentences hypotheses with a trigram language model and output the best sentence.

It is considered that the optimum language multiplier may change by the compensated acoustic scores because the method compress a dynamic range of acoustic scores. The language multiplier was, therefore, automatically optimized by utilizing the 200 best scores of preliminary experiments [61].

We evaluated the recognition results by using word error rate (WER) and keyword error rate (KER). We selected 90 keywords of player’s and place names and 227 keywords of baseball-related words, for example, “hit”, “home run”, and “stealing base”. These keywords are important for the information extraction which extracts “who did what and when” for the metadata production. In the test segments, 501 segments included these keywords and 1,030 keywords were evaluated for the KER calculation.
Table 4.1: Recognition results of clean HMM $\theta^c$ with or without the proposed compensation. Relative reduction [%] is shown in parentheses.

<table>
<thead>
<tr>
<th>Alternation</th>
<th>Confidence factor</th>
<th>WER [%]</th>
<th>KER [%]</th>
</tr>
</thead>
<tbody>
<tr>
<td>$-$ $-$</td>
<td>$-$</td>
<td>49.9</td>
<td>42.3</td>
</tr>
<tr>
<td>$+$ $-$</td>
<td>$-$</td>
<td>48.8 (0.2)</td>
<td>39.6 (6.3)</td>
</tr>
<tr>
<td>$-$ $+$</td>
<td>$+$</td>
<td>47.5 (4.8)</td>
<td>39.4 (6.7)</td>
</tr>
<tr>
<td>$+$ $+$</td>
<td>$+$</td>
<td>48.1 (3.7)</td>
<td>38.0 (10.2)</td>
</tr>
</tbody>
</table>

Table 4.2: Recognition results of noisy HMM $\theta^n$ with or without the proposed compensation. Relative reduction [%] is shown in parentheses.

<table>
<thead>
<tr>
<th>Alternation</th>
<th>Confidence factor</th>
<th>WER [%]</th>
<th>KER [%]</th>
</tr>
</thead>
<tbody>
<tr>
<td>$-$ $-$</td>
<td>$-$</td>
<td>47.5</td>
<td>40.8</td>
</tr>
<tr>
<td>$+$ $-$</td>
<td>$-$</td>
<td>47.4 (0.4)</td>
<td>36.5 (10.7)</td>
</tr>
<tr>
<td>$-$ $+$</td>
<td>$+$</td>
<td>45.8 (3.6)</td>
<td>37.0 (9.5)</td>
</tr>
<tr>
<td>$+$ $+$</td>
<td>$+$</td>
<td>46.3 (2.6)</td>
<td>35.4 (13.1)</td>
</tr>
</tbody>
</table>

4.4.2 Results

Table 4.1 shows the recognition results obtained from the clean acoustic model $\theta^c$. Improvements of the WER and KER obtained by the silence alternation and the confidence factor were compared to the baseline. In the table, $+$ means that the procedure was applied and $-$ means that the procedure was not applied. Error reduction rates relative to the result of baseline shown in row ($-$, $-$) are shown in parentheses.

The results showed that the silence alternation improved the KER and the confidence factor improved both WER and KER. The largest WER reduction of 4.8% was obtained with the confidence factor, that is to say, the result shown in the row ($-$, $+$), and the largest KER reduction of 10.2% was obtained when both procedures were adopted simultaneously.

Table 4.2 lists the recognition results of the noisy acoustic model $\theta^n$. The results showed that the alternation reduced KER and the confidence factor reduced both of WER and KER similarly to the result of the clean acoustic model $\theta^c$.

In this experiment, the difference between the training data of the silence HMM and
Table 4.3: Recognition results of $\theta_{rc}$ and $\theta_{rn}$ with or without the proposed compensation. Relative reduction [%] is shown in parentheses.

<table>
<thead>
<tr>
<th>Acoustic model</th>
<th>Alternation</th>
<th>Confidence factor</th>
<th>WER [%]</th>
<th>KER [%]</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\theta_{rc}$</td>
<td>−</td>
<td>−</td>
<td>34.3</td>
<td>25.7</td>
</tr>
<tr>
<td>$\theta_{rc}$</td>
<td>+</td>
<td>−</td>
<td>33.8 (1.5)</td>
<td>24.0 (6.9)</td>
</tr>
<tr>
<td>$\theta_{rc}$</td>
<td>−</td>
<td>+</td>
<td>33.5 (2.6)</td>
<td>24.8 (3.6)</td>
</tr>
<tr>
<td>$\theta_{rc}$</td>
<td>+</td>
<td>+</td>
<td>32.9 (4.3)</td>
<td>22.0 (14.5)</td>
</tr>
<tr>
<td>$\theta_{rn}$</td>
<td>−</td>
<td>−</td>
<td>33.1</td>
<td>23.5</td>
</tr>
<tr>
<td>$\theta_{rn}$</td>
<td>+</td>
<td>−</td>
<td>32.7 (1.1)</td>
<td>20.9 (11.0)</td>
</tr>
<tr>
<td>$\theta_{rn}$</td>
<td>−</td>
<td>+</td>
<td>32.3 (2.6)</td>
<td>21.8 (7.4)</td>
</tr>
<tr>
<td>$\theta_{rn}$</td>
<td>+</td>
<td>+</td>
<td>31.8 (4.1)</td>
<td>19.9 (15.1)</td>
</tr>
</tbody>
</table>

those of noise model $\lambda^n$ was very small because they were collected from the same broadcast sports and expected to include similar kinds of noise. In spite of that, not only the confidence factor but the alternation still reduced the KER in this experiment. These improvements showed that the proposed method successfully compensated the acoustic scores even if the acoustic model is matched to the inputs.

4.4.3 Feature space techniques for dealing with noise

This experiment examined the proposed method in conjunction with feature space techniques for noisy environments. The techniques such as the cepstrum mean normalization (CMN) described in Section 2.2.3 and spectrum subtraction (SS) described in Section 2.2.4 remove convolutive and additive noise components from input features. The proposed method, however, may degrade recognition accuracy if the noise components are precisely estimated and completely removed by these techniques. This experiment applied following two techniques, which are suitable for real-time speech recognition, for the MFCC features. One is the relative spectral processing (RASTA) [33, 34], which is a one of the CMN filter. The processing estimates the cepstrum mean from preceding input frames dynamically. This technique is applied for the MFCC in the model training stage and decoding stage. The other is the filter bank subtraction (FBS) [36], which is a one of the SS technique. The subtraction estimates the noise spectrum by using the minimum spectrum components in
Mel-frequency filter bank. This technique is applied for the feature in the decoding stage. Acoustic models, $\theta^{rc}$ and $\theta^{rn}$ corresponding to the acoustic models $\theta^{c}$ and $\theta^{n}$, were trained with RASTA filtered clean- and mixed-speech.

Table 4.3 lists the recognition results of these acoustic models, showing that the proposed method improved recognition accuracy even when the feature level techniques were used at the same time. Particularly the reduction rate of WER and KER shown in the row $(+,-,+)$. These improvements in WER and KER were statistically significant ($p < 0.05$) [62].

The experiments mentioned above obtained the confidence factor by using features whose noise components were suppressed by RASTA and FBS. However, a better confidence factor might be calculated from a feature with those noise components because these nose suppression might hinder the method from finding unconfident noisy frames. Thus, next experiment replaced the confidence factor of the condition $(\theta^{rn},-,+)$ in the Table 4.2 as that calculated from a feature without RASTA or FBS. Relative WER and KER reductions of 2.2% and 4.4% were obtained in this experiment. These reductions of WER and KER were smaller than those of 2.6% and 7.4% obtained by a confidence factor calculated with RASTA and FBS. Thus, it was concluded that a more effective confidence factor could be obtained with the matched feature even if the feature space techniques were applied.

### 4.4.4 Distribution of confidence factors

Figure 4.2 shows a histogram of the confidence factor $W^{ac}(n)$ calculated by acoustic model $\theta^{rn}$ in the test set. It is noted that about 60% of input frames gave confidence factors close to 1.0 and the acoustic scores were not changed so much. The confidence factors of the other 40% of the input frames were distributed between 0.0 and 0.95. Acoustic scores compensated to less than half of the original one were 17%.
4.5 Further investigations

This section discusses complexities of noise model $\lambda^n$ and speech model $\lambda^s$. The noise classes used in the experiments in Section 4.4 were based on the noise labels given by human. These labels were rather subjective and might not be optimal cluster for the proposed compensation. This section, therefore, introduces automatic segment clustering described in Section 3.3.1 to the noise model training. Instead of hand-labeled clusters, the noise clusters were automatically generated on the basis of GMM likelihoods from the same data used in Section 4.4. A number of noise clusters $N^{nc}$, can be regraded as the complexity of the noise models $\lambda^n$. This section examines the optimal number of noise clusters.

Moreover, it is supposed that the complexity should be balanced to improve the recognition performance of the proposed method. The speech GMMs modeling specific segments of phonemes, however, have smaller variances compared with those of noise model. This section also discuss the smaller speech models $\hat{\lambda}^s$. By reducing the number of states of the monophone HMM from 123 to 41, the speech models whose number of clusters is 41 were
examined with the complexity of the noise model.

### 4.5.1 Clustering of noise segments

It is supposed that the automatic clustering of noise segments would broaden the usefulness of this method because it obviates the need for hand-labeling the noise segments. Furthermore, the complexity of the noise model, that is to say, the number of the noise clusters, can be optimized by using the noise models $\hat{\lambda}_n^k$ generated by a segment-based iterative clustering. The clustering introduced in this section consists of two steps whose expectation step assigns each noise segment to a cluster according to the likelihood of GMMs and maximization step estimates the parameters of GMMs from the segments assigned to the clusters. The detailed procedures are followings.

1. Estimate parameters of a GMM whose number of mixtures is one from all noise segments.

2. Split any GMM which has the largest variance into two GMMs.

3. Assign each noise segment to the cluster that gives the highest GMM likelihood among all the clusters.

4. Estimate the GMM parameters of means, variances and mixture-weights with the segments assigned to the cluster.

5. Repeat steps from 2 to 4 until the desired number of clusters is obtained.

6. Increment the number of mixtures of each GMM by splitting any component that has the largest variance.

7. Estimate the GMM parameters by using the EM algorithm with the segments assigned to the cluster.

8. Repeat steps from 6 to 8. When the number of mixtures reaches the desired figure, stop clustering.
Figure 4.3: Recognition results of the automatically generated noise model $\hat{\theta}^n$ for 1 to 16 noise clusters. WER is plotted on the left-axis and error reductions relative to the result of the baseline which did not compensate the acoustic scores are plotted on the right-axis.
Figure 4.4: Recognition results of the automatically generated noise model $\tilde{\theta}^n$ for 1 to 16 noise clusters. KER is plotted on the left-axis and error reductions relative to the result of the baseline which did not compensate the acoustic scores are plotted on the right-axis.
The noise segments of 1.5 hours described in Section 4.4.1 were used for this clustering.

Figure 4.3 shows the recognition results of $\hat{\lambda}_n$ for 1 to 16 noise clusters. The acoustic model was $\theta_{rn}$ which applied RASTA and FBS. WER is plotted on the left vertical axis, and error reductions relative to the result of the baseline is plotted on the right vertical axis, where the baseline is the results yielded without the proposed compensation, that is to say, the results shown in the 5th row ($-,-$) of Table 4.3. The proposed method shown in the figure adopted both the confidence factor and the alternation for decoding. The results shown in the figure, therefore, are comparable with the results displayed in the bottom row ($+,+\)$ of the Table 4.3, that is to say, WER was 31.8%, KER was 19.9%, word error reduction was 4.1%, and keyword error reduction was 15.1%.

The results shown Figure 4.3 indicate that at least four clusters were required to obtain the best performance among these experiments. The number of clusters was not a sensitive parameter form 4 to 12. The best error reduction rate of 5.7% was obtained for four clusters.

Figure 4.4 evaluates the results in KER. The KER of 23.5% shown in the 5th row of the Table 4.3 is the baseline of the reduction rates. The plots showed similar tendencies as the WER evaluation. The best error reduction rate of 17.4% was obtained for eight clusters. This reduction was better than the reduction of 15.1% obtained by the hand-labeled clusters listed in the bottom row of Table 4.3. The best noise model consisting eight noise clusters had 192 Gaussian components in total. In order to evaluate the effectivity of the automatically generated noise clusters, a noise model consisting of single GMM was compared with the proposed eight-noise-clusters model. The single-noise-cluster model whose number of Gaussian components was increased to 192 so that it have the same number of Gaussian components with the eight-noise-clusters model. By increasing the number of the components from 24 to 192, WER of the single-noise-cluster model was reduced from 32.3% to 31.9% and KER was reduced from 22.0% to 20.9%. Compared to the result of the proposed eight-noise-clusters model, that is to say, WER was 31.1% and KER was 19.4%, the result obtained from single-noise-cluster model showed smaller improvement even with the same number of Gaussians. Followings are concluded.

- The automatic clustering of the noise model played an important role in this method.
- The hand-labeling was not good enough because training noise segments included
too many mixed noise to be classified subjectively.

4.5.2 Size of speech model

In this section, a smaller speech model $\hat{\lambda}^s$ consisting of 41 GMMs corresponding to Japanese phonemes and silence was adapted using 12.7 hours of speech of mixed condition described in Section 4.4.1. In other words, the speech GMMs were extracted from 41 states of single-state monophone HMMs. The number of the speech clusters was, therefore, reduced from 123 of $\lambda^s$ to 41 for $\hat{\lambda}^s$. This speech model also reduced the computational cost of the confidence factor calculation. Since the smaller speech model $\hat{\lambda}^s$ was considered to have larger variances than those of the larger model $\lambda^s$, the optimal number of the noise clusters $\hat{\lambda}^n$ might be different from that of $\lambda^n$.

Figure 4.5 shows the WER obtained by $\hat{\lambda}^s$ for 1 to 16 noise clusters of noise model $\hat{\lambda}^n$ on the left vertical axis. Relative error reduction rate are also displayed on the right vertical axis in the same way as in Figure 4.3. A stable improvement were obtained when the number of clusters was between 4 and 10. The best error reduction rate of 6.7% was obtained for the eight clusters. Figure 4.6 shows the KER of this experiment. The best error reduction rate of 17.9% was also obtained the eight clusters. The best result obtained from the smaller speech model $\hat{\lambda}^s$ showed better WER of 30.9% than that of the larger one $\lambda^s$ while KER was not changed. Although the larger speech model $\lambda^s$ obtained WER reductions more than 5% from three noise models consisting of 4, 6 and 8 noise clusters as shown in Figure 4.3, the smaller speech model $\hat{\lambda}^s$ obtained such reductions from five noise models consisting of 4, 6, 8, 10 and 14 as shown in Figure 4.5.

4.6 Concluding remarks

This chapter proposed a new robust speech recognition method dynamically compensating acoustic scores in the Viterbi search. The method utilizes a set of speech GMMs and a set of noise GMMs trained in advance. Their likelihoods calculated every input frame are used in two ways in this method. Firstly, the method obtains dynamic acoustic multiplier from the confidence factor as a posterior probability of speech model. Secondly, the method
CHAPTER 4. DYNAMIC COMPENSATION OF ACOUSTIC SCORES

Figure 4.5: Recognition results of the small speech model $\hat{\lambda}_s$ for 1 to 16 noise clusters $\hat{\lambda}_n$. WER is plotted in the left vertical axis and an error reduction relative to the result without compensation is plotted on the right vertical axis.
4.6. CONCLUDING REMARKS

Figure 4.6: Recognition results of the small speech model $\hat{\lambda}_s$ for 1 to 16 noise clusters $\hat{\lambda}_n$. KER is plotted in the left vertical axis and an error reduction relative to the result without compensation is plotted on the right vertical axis.
obtains an alternative likelihood for the silence HMM.

A transcription experiment implementing this method for broadcast sports commentary for the purpose of captioning and metadata extraction showed significantly lower word error rates and keyword error rates. Followings are concluded through this experiment.

- The silence alternation mainly improved KER.

- The dynamic acoustic multiplier of the confidence factor improved both WER and KER.

- The method not only improved the recognition results of the unmatched HMM trained from clean speech but also improved those of the matched HMM trained from segments including noisy speech.

- The method also improved the recognition when it was used in conjunction with feature-level noise suppression techniques such as RASTA and FBS.

- Better confidence factor could be obtained from a matched feature even if the feature space techniques were adopted.

- About 40% of the input frames were compensated by the confidence factors and 17% of the scores were compensated to less than half of the original one.

Furthermore, the complexities of the speech and noise models were discussed. The noise model generated by using automatic segment clustering performed better than the model trained with human-labeled clusters. It was also shown that the number of speech clusters could be reduced from 123 to 41 clusters without degradation. Followings are concluded thorough this discussion.

- The hand-labeling was not good enough.

- The automatic clustering of the noise model played an important role in this method.

- The smaller speech model showed more robust performance than larger one against the number of the noise clusters.
4.6. CONCLUDING REMARKS

- Overall, a 6.7% word error reduction and a 17.9% keyword error reduction were obtained by using noise models of 8 clusters and speech models of 41 clusters.

The improvement given by the proposed method should be of great advantage to the metadata extraction because many important keywords in sports commentary are uttered in very noisy environment such as a noisy crowd erupting by a homerun.

Future work will extend the dynamic acoustic multiplier to the degradation caused by speaking styles or unclear speech.
Chapter 5

Dynamic integration of multiple feature streams

5.1 Issues on integration of multiple feature streams

As described in Chapter 1, the speech recognition has not accurate enough under conditions of noisy environment or spontaneous speaking style for practical application. The method described in previous sections improves its performance by taking into account of mismatches between the acoustic model and input speech. On the other hand, it should be also addressed that recognition errors are caused by limitations of the feature space represented by joint probabilities of only several dozens of features.

It is known that our brain shrewdly integrates various cues such as amplitude and frequency modulation, so that we can recognize such unclear speech without difficulty [63]. Conventional speech recognition systems, however, are based on a joint probabilities of spectrum-oriented features described in Section 2.2.1. This chapter addresses the integration method of features representing different acoustic aspect of input speech. In the literature, many attempts integrating features have been proposed to capture such various properties of speech. These attempts are classified into three kinds of integration.

The first group analytically integrates features into a single stream. This method obtains a discriminative or de-correlated set of features by projecting feature space. The projection matrix is obtained by statistical analysis, such as principal components analysis (PCA),
linear discriminant analysis (LDA) [64], heteroscedastic discriminant analysis (HDA) [65], or maximum likelihood linear transformation (MLLT) [66]. A combined projection of HDA and MLLT is one of the most successful projection improving recognition accuracy [67]. An acoustic score of this method is a joint probability of transformed features.

The second group clusters the features into streams and then integrates the likelihoods of the feature streams statically. The acoustic score of an input frame is a weighted sum of log-likelihoods calculated from individual feature streams. In these methods, the fixed stream weights are defined in advance. Methods optimizing the stream weights have been proposed based on criteria, such as maximum likelihood (ML) [68], maximum mutual information (MMI) [69], minimum classification error (MCE) [70], or maximum entropy (ME) [71]. Stream weights optimized globally by using the ME criterion were reported to show the best performance among them [71].

The last group improves the log-likelihood integration so that the method dynamically optimizes the stream weights for each input frame. In the literature, there are various dynamic stream weights based on a stream reliability and various ways of assessing the reliability or discriminative ability of the recognition hypotheses.

For example, the $N$-best log-likelihood difference [72, 73] utilized the likelihood ratios between the best and the $N$-best hypotheses. In [72], $N$ was chosen to be five. The $N$-best log-likelihood dispersion [72, 74] captures additional $N$-best likelihood ratios which are not present in the $N$-best log-likelihood difference. These measures ensured stream reliability for audio-visual spoken digit recognition [72]. However, the discriminative ability calculated from a small number of $N$-best hypotheses tends to be underestimated in LVCSR utilizing a huge number of context dependent HMMs such as triphones because of the multiple models trained for the identical center phoneme. Furthermore, these methods require an additional sorting procedure in order to obtain the $N$-best likelihoods of hypothesized HMM states. On the other hand, following dynamic weights based on entropy do not require such additional computation and their calculation can be embedded directly in the search algorithm of the LVCSR.

The dynamic stream weights, calculated from the entropy of the search space for a given input frame, were proposed for recognizing spoken digits [75, 76]. Selective stream weights and proportional weights to the inverse entropy were proposed in [75]. They calculate the
entropy of each stream from posterior probabilities of all the HMM states defined in the
lexicon at every input frame. In the recognition of spoken digits, it is feasible to calculate
the observation probabilities of all the HMM states at every input frame because there is
a constant search space of several hundreds states. On the other hand, a real-time LVCSR
system would have to deal with an infeasible amount of likelihoods calculation to obtain
the entropy from all the HMM states. In addition to that, it may be better to take the
discriminative importance, which changes frame by frame by the beam pruning described
in Section 2.4.2, into account in a real-time LVCSR. This chapter extends the dynamic
weights to a real-time LVCSR. Following issues still remained to be addressed for the
extension of the dynamic approach.

- No additional state requiring likelihood calculation are allowed.
- The discriminative importance or difficulty should be taken into account.

Following part of this section addresses acoustic features to be integrated. As described
in Chapter 2.2, many kind of acoustic features representing different aspects of speech are
proposed in the literature. A short-term modulation feature, which is described in Sec-
tion 2.2.2 and is expected to provide different clue for the cepstrum-based features com-
plementary, is the one of the feature to be integrated with the conventional features. In
this study, we focus features extracting amplitude modulation (AM) and frequency modu-
lation of speech resonance, motivated by physiological evidence that the human auditory
system extracts such modulation information [77]. Although these modulation features are
not commonly used for speech recognition, several methods that demodulate AM and FM
signals have been proposed in the literature.

The Teager energy cepstrum coefficients (TECC) [25] are DCT coefficients of resonant
frequencies, that is to say, the center frequencies of the formants, weighted by its energy.
The resonant frequency is extracted from output signal through band pass filters by the en-
ergy separation algorithm (ESA) derived from the Teager energy operation [20,22,24]. The
comprehensive modulation spectra (CMS) [78] represents similar features to the TECC,
but it extracts resonant frequencies by using the time derivatives of the phases obtained
through orthogonal band-pass filters. On the other hand, the frequency modulation per-
centage (FMP) [26, 27] was proposed to capture fluctuations in speech resonances. The
5.2. INTEGRATION OF MULTIPLE STREAMS

feature is the ratio of the second moment over the first one of instantaneous frequency extracted by the ESA. Amplitude modulation also can be captured in a similar manner.

Psychophysical experiments also showed that these modulation features provide auxiliary cues to conventional spectrum-oriented features [21]. These modulation features are, therefore, often joined with the cepstrum features [27] because accurate results can not be obtained from the modulation features alone. This chapter also addresses the limitations of improvements obtained by these joint feature streams or acoustic scores calculated as a joint probability of the features.

The rest of this paper is organized as follows. Section 5.2 presents an outline of the proposed method and details of its stream weighting, and Section 5.3 describes the features we used. Experimental results on broadcast news are given in Section 5.4. Section 5.5 discusses the issues related to the features, differences in dynamic weights, and implementation.

5.2 Integration of multiple streams

This section present a novel method of integrating likelihoods of multiple feature streams representing different acoustic aspects for robust speech recognition. The integration algorithm dynamically calculates a frame-wise stream weight so that a higher weight is given to a stream that is robust to a variety of noisy environments or speaking styles. Such a robust stream is expected to show discriminative ability. The weight is calculated in real-time from the entropy reduction of a search space, that is to say, active HMM states, caused by observing each input frame. The reduction can be obtained without any additional calculation of state probabilities. Furthermore, the weight takes the width of the search space into account by calculating the marginal entropy from the number of active states.

Figure 5.1 shows the outline of the proposed stream integration. The proposed LVCSR decoder integrates various feature streams, such as energy, AM, and FM, as described in Section 5.3. A frame-wise entropy of a stream is calculated from the likelihoods of active states in hypotheses generated in the search process. The number of active states also yields the marginal entropy, which is the entropy of search space before observing an input feature, and then an entropy reduction caused by an observed feature is calculated by
Figure 5.1: Outline of the proposed dynamic stream weighting.
subtracting the entropy yielded after observing an feature from the marginal entropy. The method dynamically yields a log-linear stream weight proportional to the entropy reduction in order to give a higher weight to a discriminative stream or a stream giving more information. The likelihoods of the multiple features are integrated into an acoustic score in a log-linear domain. The Viterbi search is executed while changing the stream weights frame by frame.

5.2.1 Stream weights based on mutual information

As shown in Figure 5.1, the proposed method gives higher weight to a stream \( k \in \{1 \ldots N_{\text{strm}}\} \) with larger mutual information, given observation vector \( o_k(n) \), where \( n \) is a frame index, \( k \) is stream index and \( N_{\text{strm}} \) is number of streams to be integrated. The stream given a higher weight is supposed to be less affected than the others by background noise or spontaneous speech. The entropy reduction of the search space \( I_k(n) \) given by \( o_k(n) \) is calculated as follows,

\[
I_k(n) = H^0(n) - H_k(n),
\]

(5.1)

where \( H^0(n) \) is the marginal entropy of search space \( \Lambda(n) \), that is to say, the entropy before observing a feature vector, and \( \Lambda(n) \) is a set of active HMM states \( \lambda \in \Lambda(n) \) at frame \( n \). \( H_k(n) \) is the entropy of \( \Lambda(n) \) given an observation vector \( o_k(n) \). Equation 5.1 calculating the entropy reduction is a similar formula to the calculation of the mutual information. The proposed method, however, utilizes \( H_k(n) \) instead of the conditional entropy so as to capture the information yielded by \( o_k(n) \).

The proposed method, in another sense, maps the entropy \( H_k(n) \) to a non-negative value of stream reliability by using a linear monotonic decreasing function. In this sense, the marginal entropy \( H^0(n) \) is utilized as an offset of the linear transformation so that the reliability will be non-negative. The offset can not be determined in advance in the method yielding the entropy from only active HMM states \( \Lambda(n) \), because the dynamic range or the maximum value of the entropy is not constant but dependent on the frame \( n \) or number of active states \( N_{\text{active}}(n) \). Therefore, we assume the prior probabilities of \( \lambda \in \Lambda(n) \) to be a uniform distribution to ensure that the reliability is a non-negative value. Consequently,
\(H^0(n)\) is only dependent on \(N^{active}(n)\) and is calculated as

\[
H^0(n) = \log N^{active}(n)
\] (5.2)

The forward Viterbi path scores in \(\Lambda(n)\) may be used for the calculation of the entropy reduction. However, in such a case, the difference between the entropies calculated by the forward Viterbi scores before and after observing \(o_k(n)\) does not provide a measure of the discriminative ability yielded by \(o_k(n)\). Because \(I_k(n)\) is not mutual information but entropy reduction as defined in Equation 5.1, it is not always positive for any observation \(o_k(n)\). Hence, the forward Viterbi path scores can not be used in our method since they could give a negative \(I_k(n)\) even for a discriminatively important input frame.

The entropy given by stream \(o_k(n)\) can be written as follows,

\[
H_k(n) = \sum_{\lambda \in \Lambda(n)} P(\lambda|o_k(n)) \log P(\lambda|o_k(n)) ,
\] (5.3)

where we define a posterior probability \(P(\lambda|o_k(n))\) given \(o_k(n)\) as follows,

\[
P(\lambda|o_k(n)) = \frac{\tilde{P}(o_k(n)|\lambda)}{\sum_{\lambda_j \in \Lambda(n)} \tilde{P}(o_k(n)|\lambda_j)} ,
\] (5.4)

assuming

\[
P(\lambda_1) = \cdots = P(\lambda_{N^{active}(n)}) = \frac{1}{N^{active}(n)}
\] (5.5)

and

\[
P(o_k(n)) = \frac{1}{N^{active}(n)} \sum_{\lambda \in \Lambda(n)} \tilde{P}(o_k(n)|\lambda) .
\] (5.6)

As integrated streams have potentially different discriminative abilities, their likelihoods should be balanced with each other so that the contribution of a stream which degrades the recognition performance will be underestimated. Therefore, a statically weighted observation probability \(\tilde{P}(o_k(n)|\lambda)\) is defined to compensate the discriminative ability of stream \(k\):

\[
\tilde{P}(o_k(n)|\lambda) = P(o_k(n)|\lambda)^{w_k} ,
\] (5.7)
where $w_k$ is the static weight of the stream $k$ and $P(o_k(n)|\lambda)$ is the observation probability.

As described in Section 5.1, these static weights can be optimized by ME or MCE using training data in advance. The proposed dynamic stream weights described in the following are calculated from these compensated likelihoods.

The proposed integration multiplies each compensated likelihood in a log-linear domain by the dynamic weight $W_{k \text{strm}}(n)$, and then obtains the acoustic score by summing the weighted log-likelihoods.

$$\log P(o_k(n)|\lambda) = \sum_{k=1}^{N_{\text{strm}}} W_{k \text{strm}}(n) \log \hat{P}(o_k(n)|\lambda) \tag{5.8}$$

Finally, the forward Viterbi search is executed using the following recursive equation,

$$\alpha^{\text{Viterbi}}_j(n+1) = \max_i \alpha^{\text{Viterbi}}_i(n) a_{ij} P(o(n)|\lambda_j) \tag{5.9}$$

where $\alpha^{\text{Viterbi}}_j(n)$ is an accumulated Viterbi score of state $j$ at frame $n$ on the Viterbi path and $a_{ij}$ is the transition probability from state $j$ to $i$.

Below, we compare the proposed dynamic weights $W_{k \text{MI}}(n)$ based on the entropy reduction $I_k(n)$ with two conventional dynamic weights, $W_{k \text{MinH}}(n)$ and $W_{k \text{IncH}}(n)$ based on the entropy $H_k(n)$.

### 5.2.2 Selective weighting (MinH)

This integration uses selective weights [75]; therefore, the selected stream is the one giving the minimum entropy. The method obtains the stream weight $W_{k \text{MinH}}(n)$ as follows:

$$W_{k \text{MinH}}(n) = \begin{cases} 
1.0 & \text{if } k = \arg \min_k H_k(n) \\
0.0 & \text{otherwise.} 
\end{cases} \tag{5.10}$$
5.2.3 **Inverse entropy (InvH)**

This integration introduces inverse entropy [75] in order to obtain a higher weight for a stream giving lower entropy. The stream weight $W_{k}^{InvH}(n)$ is calculated as follows:

$$W_{k}^{InvH}(n) = \frac{1}{H_{k}(n)} \sum_{l=1}^{N_{strm}} \frac{1}{H_{l}(n)}$$  \hspace{1cm} (5.11)

5.2.4 **Entropy reduction (MI)**

The proposed integration obtains the dynamic weights from the entropy reduction described in Equation (5.1), which takes into account the dynamics of the number of active states $N^{active}(n)$ in the marginal entropy $H^{0}(n)$. Normalizing the mutual information $I_{k}(n)$, the method calculates the weight $W_{k}^{MI}(n)$ in a manner similar to the calculation of $W_{k}^{InvH}(n)$,

$$W_{k}^{MI} = \frac{I_{k}(n)}{\sum_{l=1}^{N_{strm}} I_{l}(n)}.$$  \hspace{1cm} (5.12)

5.3 **Integrated streams**

In order to examine the proposed stream integration, we pick up three streams based on an auditory filter-bank, as shown in Figure 5.2. The orthogonal auditory filter-bank is implemented with asymmetrical gamma-tone filters [79] whose impulse response is given by

$$G^{<}_r(t) = Af_r^3 t^3 \exp \left( -2\pi \cdot 1.019 \cdot \text{ERB}(f_r) t \right) \cos(2\pi f_r t),$$  \hspace{1cm} (5.13)

and

$$G^{>}_r(t) = Af_r^3 t^3 \exp \left( -2\pi \cdot 1.019 \cdot \text{ERB}(f_r) t \right) \sin(2\pi f_r t),$$  \hspace{1cm} (5.14)

where $t$ is the sample index, $r$ is the filter index, $A$ is the gain parameter, $f_r$ is the center frequency of filter $r$, and $\text{ERB}(\cdot)$ is a function giving the equivalent rectangular bandwidth. The filter yields complex output $y_r(t)$; the filter $G^{<}_r(t)$ yields the real part of a band passed signal $y_r^{<}(t)$ and the filter $G^{>}_r(t)$ yields the imaginary part $y_r^{>}(t)$. These filters can be efficiently implemented with the second order infinite impulse response (IIR) filters [80]. The method compensates the length of the filter response by shifting the output $y_r(t)$ with
5.3. INTEGRATED STREAMS

Figure 5.2: Integrated streams.
Differences in the length of the impulse responses are compensated by shifting a time

\[ T^G(r) = \arg \max_t \left((G_r^1(t))^2 + (G_r^2(t))^2\right). \quad (5.15) \]

The output signal \( y_r(t) \) of the \( r \)-th filter is then decomposed into its instantaneous amplitude \( A_r(t) \) and instantaneous phase \( \phi_r(t) \) as follows,

\[ A_r^2(t) = |y_r(t)|^2 \quad (5.16) \]
\[ \phi_r(t) = \angle y_r(t) \quad (5.17) \]

Three kinds of features are calculated from \( A_r^2(t) \) and \( \phi_r(t) \). The integrated streams are derived from these features through the logarithm and the DCT calculations. Each stream consists of 39 parameters: 12 cepstrum coefficients (without C0) with log energy and their first- and second-order regression coefficients (\( \Delta + \Delta\Delta \)). Descriptions of these features are given below.
5.3. INTEGRATED STREAMS

Figure 5.4: Feature of AM calculated from Japanese speech of “m a ch i n i m a Q t a”.

Figure 5.5: Feature of FM calculated from Japanese speech of “m a ch i n i m a Q t a”.
5.3.1 Energy

The first stream captures the expectations of filter-bank energy at frame $n$. This stream is supposed to have the same aspect as mel-frequency cepstrum coefficients (MFCC) described in Section 2.2.1. The $r$-th element of the feature $energy_r(n)$ is calculated from the amplitude $A_r(t)$ as follows,

$$energy_r(n) = E_n \left[ A_r^2(t) \right] = \frac{1}{T_w} \sum_{t=1}^{T_w} A_r^2(T^s \cdot n + t), \quad (5.18)$$

where $T_w$ is frame width and $T^s$ is a frame shift. The features calculated from Japanese speech “m a c h i n i m a Q t a” is shown in Figure 5.3.

5.3.2 Amplitude modulation

The second feature captures the time derivative of the energy. We focus on the ratio of the energy differential, which is calculated as

$$\dot{A}_r^2(t) = A_r^2(t) - A_r^2(t-1), \quad (5.19)$$

and energy $A_r^2(t)$ so that Weber’s law of just noticeable difference (JND) is taken into account. An element of the feature $AM_r(n)$ is calculated by summing the ratio over a frame as follows:

$$AM_r(n) = \left| E_n \left[ \frac{\dot{A}_r^2(t)}{A_r^2(t)} \right] \right| = \left| \frac{1}{T_w} \sum_{t=1}^{T_w} \frac{\dot{A}_r^2(T^s \cdot n + t)}{A_r^2(T^s \cdot n + t)} \right|. \quad (5.20)$$

This feature captures the energy derivative calculated form just one frame. The modulation frequencies captured by this feature are different from the “energy” regression coefficients, which are calculated from five adjacent frames as the smoothed derivative. This feature should yield an additional clue to the conventional spectrum-oriented feature, since it emphasizes the raising and falling edges of the input speech (Figure 5.4).
5.3.3 Frequency modulation

The third feature captures time derivative of resonant frequency. This frequency modulation feature has an extraction algorithm similar to FMP feature extraction, but it extracts the drift of the resonant frequency.

Assuming that instantaneous frequency $\dot{\phi}_r(t)$ is extracted by

$$\dot{\phi}_r(t) = \phi_r(t) - \phi_r(t - 1), \quad (5.21)$$

the feature extracts the resonant drift $\ddot{\phi}_r(t)$ as

$$\ddot{\phi}_r(t) = \dot{\phi}_r(t) - \dot{\phi}_r(t - 1). \quad (5.22)$$

Moreover, Weber’s ratio of the resonant drift and the resonant frequency $\ddot{\phi}_r(t)/\dot{\phi}_r(t)$ is weighted by the energy $A_r^2(t)$ in order to prevent the ratio from being influenced by non-speech resonants. An element of the feature $FM_r(n)$ is the weighted sum of the ratios in a frame.

$$FM_r(n) = \left| E \left[ A_r^2(t) \frac{\ddot{\phi}_r(t)}{\dot{\phi}_r(t)} \right] \right| = \left| \sum_{t=1}^{T_s} A_r^2(T_s \cdot n + t) \frac{\ddot{\phi}_r(T_s \cdot n + t)}{\dot{\phi}_r(T_s \cdot n + t)} \right| \quad (5.23)$$

As shown in Figure 5.5, the resonant movement is emphasized in this feature. Although it may be difficult to obtain a good recognition result from these modulation features alone, these additional clues are expected to help the recognition by integrating with the conventional stream.

5.4 Experiments

5.4.1 Setup

Experiments were performed to examine the recognition performance of the proposed stream integration. The evaluation speech data was extracted from NHK’s Japanese broadcast news. The 16 most difficult news topics were selected from the programs aired in July.
2004. The data consisted of read speech in quiet studios, noisy field reports, and conversational commentaries uttered by male announcers and reporters; there were 643 segments comprising 8,391 words “all”. Subsets of “field” reports (395 segments comprising 4,905 words) and “spontaneous” commentaries (231 segments comprising 3,220 words) were also evaluated. There were 97 common segments in these subsets, and the complement set of the union of these subsets included 114 segments of read speech in a quiet studio.

Triphone HMMs with 16-mixtures and about 4K clustered states were trained from about 100 hours of NHK’s multi-conditioned news data uttered by male speakers. All the features described in this experiment were extracted from pre-emphasized speeches digitized at 16 kHz and 16 bits. The features were obtained from band-limited signals passed through 40 channels of gamma-tone filters with a frame of width \( T_w = 25\text{ms} \) width and shift \( T_s = 10\text{ms} \) shift. Each stream element was normalized to have zero mean and unit variance on the training data.

The n-gram language models used in this experiment were word bigrams for the first pass in the continuous speech recognition and word trigams for the second pass. The training texts were NHK’s Japanese news manuscripts consisting of 127M words extending back over 10 years. As more recent news had a higher probability of frequently appearing, the n-gram language models were trained with a higher weighting factor to the latest news in an n-gram count level [1]. There were 61K vocabulary words, and the trigram language model showed a perplexity of 21.9, with an out-of-vocabulary rate of 0.4% against the evaluation data.

In order to compare the recognition results obtained from the streams yielding different dynamic range of acoustic score, the acoustic multiplier described in Section 2.4.3 was optimized so as to minimize word errors, while the other search parameters of the beam width and the language multiplier were fixed. In this optimization, total word errors of insertion, substitution and deletion were taken into account. The fixed search parameters were listed in the Table 5.1.
5.4. EXPERIMENTS

Table 5.1: Search parameters used in this experiment.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Viterbi score beam in natural log scale</td>
<td>200</td>
</tr>
<tr>
<td>Maximum HMM nodes being active</td>
<td>2,000</td>
</tr>
<tr>
<td>Language multiplier</td>
<td>10</td>
</tr>
<tr>
<td>Insertion penalty</td>
<td>0</td>
</tr>
</tbody>
</table>

Table 5.2: Recognition results of individual feature-streams.

<table>
<thead>
<tr>
<th>Feature stream</th>
<th>Word error rate [%]</th>
<th>Acoustic multiplier</th>
<th>Number of active stats</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>all</td>
<td>field</td>
<td>spontaneous</td>
</tr>
<tr>
<td>Energy [39]</td>
<td>7.4 (6.5)</td>
<td>7.6 (6.7)</td>
<td>12.7 (11.1)</td>
</tr>
<tr>
<td>AM [39]</td>
<td>10.9 (9.3)</td>
<td>10.6 (9.2)</td>
<td>18.2 (15.5)</td>
</tr>
<tr>
<td>FM [39]</td>
<td>11.6 (9.8)</td>
<td>12.0 (10.3)</td>
<td>18.5 (15.4)</td>
</tr>
</tbody>
</table>

Dimension of the feature vector is shown in square brackets. WER discarding insertion errors is shown in parentheses.

5.4.2 Single stream

Table 5.2 shows the word error rate (WER) of the recognition results of each individual feature stream. The WER discarding insertion errors, that is to say, the WER calculated from only substitution and deletion errors, are shown in parentheses. The reason for omitting insertion errors was the difficulty of the unbiased evaluation of many repetitions consisting of several words. It is difficult to define unbiased appropriate references for these repetitions, which contain disfluencies or lack of several phonemes, and the repetitions are also manually deleted whether these words are recognized correctly or not in the practical use of the LVCSR for closed-captioning service [3]. The correction cost of these repetitions is negligible because they are correctable without any manual keyboard input following the recognition [3]. The reference used in the experiments, therefore, discarded such repetitions, which were mainly contained in the “spontaneous” subset.

Table 5.2 also shows the optimized acoustic multiplier and frame-averaged number of active states. Among the individual streams of “Energy”, “AM” and “FM”, the modulation features, “AM” and “FM”, produced larger WERs than the fine result obtained from
Table 5.3: Recognition results of joint feature streams.

<table>
<thead>
<tr>
<th>Feature stream</th>
<th>Word error rate [%]</th>
<th>Acoustic multiplier</th>
<th>Number of active states</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>all</td>
<td>field</td>
<td>spontaneous</td>
</tr>
<tr>
<td>EAF [111]</td>
<td>8.9 (7.4)</td>
<td>9.0 (7.8)</td>
<td>14.8 (12.5)</td>
</tr>
<tr>
<td>HLDA [39]</td>
<td>8.2 (7.1)</td>
<td>8.8 (7.8)</td>
<td>12.3 (10.6)</td>
</tr>
<tr>
<td>HLDA [57]</td>
<td>7.5 (6.5)</td>
<td>7.8 (6.9)</td>
<td>12.3 (10.3)</td>
</tr>
</tbody>
</table>

Dimension of the feature vector is shown in square brackets.
WER discarding insertion errors is shown in parentheses.

“Energy” stream. The optimized acoustic multipliers were nearly 1.0 at all streams and frame-averaged number of active states did not show large differences. The equality of the search width of each experiments is confirmed.

Table 5.3 compare the recognition results of streams joined with tree features. The table also shows the dimension of the feature vectors in square brackets. “EAF” shows a result of a 111-dimensional stream joined with three kind of DCT coefficients and log-power (12[coefficients] × 3 + 1[log-power]) with their regressions (37 × 3). “HLDA” shows a result of feature stream transformed by HLDA-MLLT [67] matrix reducing 111 dimensions to a 39 or 57-dimensional stream and de-correlating each element of the stream. The HLDA-MLLT matrix was trained with the same data as used in HMM training. The dimension of the transformed feature “HLDA[57]” was optimized by the Fibonacci search method [81]. The results show that the “EAF[111]” degraded WERs compared with the best result obtained from the individual stream, that is to say, “Energy[39]” in Table 5.2. In addition, “HLDA [57]” could not improve the WERs except under the condition of “spontaneous” speech.

“EAF[111]” yielded a smaller optimized acoustic multiplier of 0.4, while the other features yielded the multiplier of nearly 1.0, because its increased number of feature dimensions, [111], would broaden its dynamic range of the likelihoods. The frame-averaged number of active states did not show large differences too.
Table 5.4: Recognition results of stream integration ($\forall k, w_k = 1.0$).

<table>
<thead>
<tr>
<th>Integration $W_k^{strm}(n)$</th>
<th>Word error rate [%]</th>
<th>Acoustic multiplier</th>
<th>Number of active states</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>all</td>
<td>field</td>
<td>spontaneous</td>
</tr>
<tr>
<td>Static</td>
<td>7.9 (6.6)</td>
<td>7.9 (6.8)</td>
<td>13.7 (11.4)</td>
</tr>
<tr>
<td>MinH</td>
<td>7.5 (6.3)</td>
<td>7.2 (6.3)</td>
<td>13.0 (11.0)</td>
</tr>
<tr>
<td>InvH</td>
<td>7.6 (6.3)</td>
<td>7.5 (6.5)</td>
<td>13.0 (11.0)</td>
</tr>
<tr>
<td>MI</td>
<td>7.8 (6.4)</td>
<td>7.6 (6.5)</td>
<td>13.5 (11.1)</td>
</tr>
</tbody>
</table>

WER discarding insertion errors is shown in parentheses.

5.4.3 Integrated likelihoods

This section describes recognition results of integrations based on the likelihoods. All the acoustic models integrated in these experiments were composed so as to retain the identical structure of the state-tying and probability matrices of the state transitions. Each HMM was generated from the reference HMM of the 111-dimensional stream “EAF[111]”, to which three kinds of the features were joined. The parameters of the probability density functions of each state were re-estimated by using each feature stream while keeping the state-tying structure and state transition matrices. These identical structure and transition probabilities accomplished the Viterbi search on a single pre-compiled state network with varying the stream weight.

Table 5.4 shows the results of the stream integration. In this experiment, the static stream weight $w_k$ described in Equation (5.7) was unit weight for all stream $k$. “Static” shows the result of static weighting, that is to say, the integration did not use dynamic weights and $W_k^{strm}(n)$ was 1.0 for all $k$ and $n$. It had higher word error rates than those of “Energy[39]”. “MinH”, “InvH”, and “MI” indicate the dynamic weighting described in Section 5.2.1. These results showed that the dynamic weights did not reduce the total WERs (“all”) compared with the result obtained by “Energy[39]”. “MinH”, however, reduced the WER of the “field” condition. The dynamic weights also reduced the WER discarding insertion errors under the “field” condition. These results were contrastive to those of “HLDA[57]”, which showed the best result among the joined feature shown in Table 5.3.
Table 5.5: Recognition results of stream integrations using optimized static stream-weights $w_k$ of $0.51:0.28:0.21$ (Energy:AM:FM).

<table>
<thead>
<tr>
<th>Integration $W^\text{strm}(n)$</th>
<th>Word error rate [%]</th>
<th>Acoustic multiplier</th>
<th>Number of active states</th>
</tr>
</thead>
<tbody>
<tr>
<td>Static</td>
<td>7.6 (6.5) 7.5 (6.5) 13.3 (11.3)</td>
<td>1.0</td>
<td>2291</td>
</tr>
<tr>
<td>MinH</td>
<td>7.3 (6.2) 7.0 (6.1) 12.2 (10.5)</td>
<td>1.6</td>
<td>2340</td>
</tr>
<tr>
<td>InvH</td>
<td>7.5 (6.3) 7.4 (6.4) 12.5 (10.6)</td>
<td>2.5</td>
<td>2257</td>
</tr>
<tr>
<td>IEWAT</td>
<td>7.3 (6.2) 7.0 (6.1) 12.3 (10.7)</td>
<td>1.6</td>
<td>2344</td>
</tr>
<tr>
<td>MI</td>
<td>7.0 (5.9) 6.9 (6.0) 12.1 (10.3)</td>
<td>2.3</td>
<td>2240</td>
</tr>
</tbody>
</table>

WER discarding insertion errors is shown in parentheses.

The selective method “MinH” yielded percentages of frames selected to use the energy, $AM$, and $FM$ features of 70%, 16%, and 15%, respectively. This showed that the energy feature was the most reliable among them and consistent with the result shown in Table 5.2.

In a similar manner, the average weights of each stream of energy, $AM$, and $FM$ were calculated over the evaluation frames. “InvH” yielded average weight ratios of 0.45:0.28:0.27, and “MI” yielded 0.37:0.31:0.32. “MI” degraded the WERs more than “InvH” because of its narrower weight dynamics.

It is considered that the “MI” did not improve the WERs compared to the “MinH” or “InvH” due to its narrower dynamics of the weights. Through these experiments, the acoustic multiplier was optimized to nearly 1.0 except for the “Static” condition; therefore, dynamic range of the acoustic scores were not changed so much by these dynamic weight.

As described in Section 5.1, the static stream weights $w_k$ can be optimized in advance by using training data. Table 5.5 shows the results of the dynamic stream integrations in conjunction with the optimized static stream weights. The optimization of the static stream weights was based on the maximum entropy criterion [69, 71], and yields the weights $w_k$ maximizing the posterior log-likelihood $\log P(\hat{Q}|O)$ given training feature vectors $O = \{o(1) \cdot o(N^o)\}$ and corresponding state sequence $\hat{Q}$.

$$
\log P(Q|O) = \sum_{n=1}^{N^o} \left\{ \sum_k w_k \log P(\hat{q}(n)|o_k(n)) - \sum_j \sum_k w_k P(q_j|o_k(n)) \right\}. 
(5.24)
$$
5.4. EXPERIMENTS

The state clusters \( \bar{q} \) of 42 Japanese phonemes were taken into account to make use of the optimization for the LVCSR task processing thousands of states. Namely, the posterior probability \( P(q|o_k(t)) \) in Equation (5.24) were replaced by

\[
\bar{P}(\bar{q}|o_k(n)) = \max_{q \in \bar{q}} P(q|o_k(n)).
\] (5.25)

Optimization on a 9-hour subset of the training data yielded the static weights of 0.51, 0.28 and 0.21 for energy, AM and FM.

It was noted that the optimized acoustic multipliers were larger than the multipliers shown in Table 5.4 because the optimized static stream weight \( w_k \) changed the dynamic range of the acoustic scores.

The table also shows results of the “IEWAT” method [75], “IEWAT”, that is to say, inverse entropy weighting with average entropy at each frame level threshold. It obtained the largest improvement in experiments on recognition of connected digits in telephone speech, which integrated the output of artificial neural networks trained for the full combinations of multi-stream [75, 82] In this weighting scheme, the average entropy of all the streams for a frame is calculated by,

\[
H(n) = \frac{1}{N_{strm}} \sum_{k=1}^{N_{strm}} H_k(n)
\] (5.26)

The average entropy is used as a dynamic threshold for the frame. The likelihoods of the streams having an entropy greater than the threshold are weighted much less, whereas the likelihoods of the streams having an entropy lower than the threshold are weighted inversely proportionally to their respective entropies. The stream weight \( W_k^{IEWAT}(n) \) is calculated as follows:

\[
\hat{H}_k(n) = \begin{cases} 10000 & : H_k(n) > H(n) \\ H_k(n) & : H_k(n) \leq H(n) \end{cases}
\] (5.27)

\[
W_k^{IEWAT} = \frac{1}{\sum_{j=1}^{N_{strm}} 1/\hat{H}_j(n)}
\] (5.28)

The selective method “MinH” yielded the percentages of frames selected to use energy, AM and FM features of 99.8%, 0.2% and 0.1%, respectively, and “InvH”, “IEWAT”, and “MI” yielded ratios of average weights to the evaluation data of 0.49:0.26:0.24, 0.99:0.008:0.001,
Table 5.6: Recognition results of the dynamic stream weights calculated by the whole states.

<table>
<thead>
<tr>
<th>Integration $W_{k}^{\text{strm}}(n)$</th>
<th>Word error rate [%]</th>
<th>Acoustic multiplier</th>
<th>Number of active states</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>all</td>
<td>field</td>
<td>spontaneous</td>
</tr>
<tr>
<td>MinH.full</td>
<td>7.3 (6.2)</td>
<td>7.0 (6.1)</td>
<td>12.2 (10.5)</td>
</tr>
<tr>
<td>Inv.full</td>
<td>7.4 (6.2)</td>
<td>7.2 (6.3)</td>
<td>12.6 (10.6)</td>
</tr>
<tr>
<td>MI.full</td>
<td>7.0 (5.9)</td>
<td>6.8 (5.9)</td>
<td>12.2 (10.3)</td>
</tr>
</tbody>
</table>

WER discarding insertion errors is shown in parentheses.

and 0.61:0.23:0.15. These results showed that the optimized static stream weights significantly reduced WERs of “MI” compared with the results shown in Table 5.4. Dynamic integration improved the WERs in both “field” and “spontaneous” subsets, while the static integration of “HLDA” improved the WER only in the “spontaneous” subset and “Static” improved the WER only in the “field” subset.

Finally, the proposed dynamic weight “MI” based on the entropy reduction reduced WER by 4.1% relative to “MinH” which showed the best result from among the conventional stream integration. The proposed method also obtained WER reductions of 5.4%, 9.2%, and 4.7% relative to the single stream of “Energy” for the evaluation set of “all”, “field”, and “spontaneous”, respectively.

The WER reduction of the proposed integration (MI) from the WER obtained for the single stream (Energy) for all evaluation data (all) was significant in a paired difference t-test [83] at a significance level of 0.05, but WER reduction of the conventional integration (MinH) was not significant. The difference in WERs between the proposed integration (MI 7.0% (5.9%)) and the conventional integration (MinH 7.3% (6.2%)) was not significant, but the difference in WERs discarding insertion errors showed significant improvement.

For the dynamic weight of $W_{k}^{ MI}(n)$ calculated directly from the uncompensated likelihood $P(o_k(n)|\lambda)$, (i.e., $\forall k, w_k = 1.0$ were used in the calculation of $W_{k}^{ MI}(n)$ and optimized $w_k$ were used in the Viterbi search) the WERs shown in Table 5.5 (“MI”) were degraded to 7.8%, 7.3%, and 13.7% for “all”, “field”, and “spontaneous”, respectively. This result suggests the importance of the compensated likelihood $\tilde{P}(o_k(n)|\lambda)$ in the estimation of the dynamic weight.
Table 5.6 shows the results of stream integration calculating the dynamic weights for all the 3,972 states of the HMM set. The differences in WERs between the results shown in Table 5.5 and those of Table 5.6 were very small, although the proposed method required only 56% of the computational cost relative to the experiment calculating all the 3,972-state probabilities.

5.5 Discussion

This section discusses issues related to the features used in this experiment and the implementation of the proposed method. Because the AM and the FM features were very weak compared with the energy feature, good results were not obtained from the conventional static integrations of the joint feature “EAF”, the transformed feature “HLDA” or the static stream weight “Static”. Though carefully designed features may improve the recognition performance of these integration schemes, the major advantage of the proposed method was the experimental fact that the dynamic weights improved the WERs by integrating these weak streams, that is to say, the AM stream and the FM stream. It is considered that the weak streams improved the WERs owing to their local robustness against particular kinds of environmental conditions or speaking styles. Therefore, a stream robust to a certain kind of noise for specific phonemes has a chance to improve the WER by the proposed method, even if the noise is unseen in the HMM training. Of course, the proposed method would perform better if it integrated more accurate features or features whose vector components are optimized to represent complementary speech aspects.

Figure 5.6 compares the proposed dynamic weights \( W^{MI}(n) \) and the conventional dynamic weights \( W^{InvH}(n) \). The top graph shows the envelope of an input waveform. The second graph shows the number of active states for the same time period as the top graph together with the recognition results for \( W^{InvH}(n) \) and \( W^{MI}(n) \). \( W^{InvH}(n) \) did not improve the WER of the speech segment shown in Figure 5.6 in comparison with the result of “Energy”. The WER of \( W^{InvH}(n) \) shown in “E:” was 50%. \( W^{MI}(n) \) improved the WER of the segment and the WER of \( W^{MI}(n) \) shown in “C:” was 0%. The third and the bottom graphs show the dynamic weights \( W^{kMI}(n) \) and \( W^{kInvH}(n) \) for the streams of “Energy”, “AM”, and “FM. \( W^{MinH}(n) \) selected a stream yielding the maximum \( W^{MI}(n) \); that is to
Figure 5.6: Comparison of proposed dynamic weights $W_k^{MI}(n)$ and conventional dynamic weights $W_k^{InvH}(n)$. The top graph shows the envelope of an input waveform. The second graph shows the number of active states in the log scale together with recognition results. The WER of the result shown in E:, which is the result for “Energy” and $W_k^{InvH}(n)$, was 50%. The WER of the result shown in C:, which is the result of $W_k^{MI}(n)$, was 0%. The third and bottom graphs show $W_k^{MI}(n)$ and $W_k^{InvH}(n)$ for the same time period as the top graph.
say, $W^{MinH}(n)$ selected “Energy” for the all frames. Fairly smooth dynamic weight trajectories were already yielded by both methods without using the entropies of neighboring frames. As described in Section 5.2, $W^{MI}(n)$ maps the entropy $H_k(n)$ to a non-negative reliability by using linear monotonic decreasing function, while $W^{InvH}(n)$ maps it using the inverse function. $W^{MinH}(n)$ utilizes a more non-linear function than $W^{InvH}(n)$. The difference in $W^{InvH}(n)$ between “AM” and “FM” was smaller than the difference in $W^{MI}(n)$. It is considered that the nonlinear characteristic of the inverse function did not capture the difference in the reliabilities of these streams, whereas the proposed method yielded better estimation of reliabilities of these stream by using the linear function and the static compensation weight $w_k$ than the conventional method.

The other issue is computational cost when implementing the proposed method on a real-time system. As mentioned above, the proposed method, which takes the active states into account, does not increase the number of states requiring a likelihood calculation. The number of likelihood calculations, however, increases with the number of streams; for example, the proposed integration under the experimental conditions than those of a single stream such as “Energy”. In this case, the pool of state-likelihoods, from which the method obtained the dynamic weights, can help the implementation of the proposed method. To obtain the entropy of the active states, the proposed method pools the all likelihoods of the active states before updating Viterbi score described in Equation(5.9). This likelihood pool can be compute in parallel although the Viterbi updates can not computed in parallel because of their merging process of the Viterbi scores. In this method, the process of the Viterbi update can be implemented to compute the scores by referring to the pooled likelihood after pooling. In addition, a recent multi-core processor makes the parallelization easy.

5.6 Concluding remarks

This chapter proposed a new method of real-time stream integration utilizing mutual entropy reduction given by input feature vectors in the search space, that is to say, active HMM states of hypotheses. The method calculates frame-wise dynamic stream weights from the entropy reduction and dynamically integrates the likelihoods of multiple feature
CHAPTER 5. DYNAMIC INTEGRATION OF MULTIPLE FEATURE STREAMS

Motivated by physiological evidence, we integrated three streams extracting energy, AM, and FM.

Transcription experiments implementing this method for Japanese broadcast news were performed for the purpose of automatic captioning. The results showed as follows,

- The proposed dynamic weight reduced error words by 5.4% relative to the result of the single stream of “Energy” feature.

- A combination of the proposed integration with the static stream weights optimized by ME reduced error words by 4.1% relative to the conventional dynamic integration based on the minimum entropy (“MinH”).

- The proposed integration improved the recognition accuracy while the joint features could not improve it even if it was transformed discriminatively.

- If the static stream weights were not optimized, the proposed integration did not improve the recognition accuracy compared to the conventional method of the minimum entropy (“MinH”) or the inverse entropy (“InvH”) due to its narrower dynamics of the weights.

- The optimization of the static stream weight significantly reduced word error rate of the proposed method.

- Difference in word error rates were extremely small if proposed method estimated the entropy from all the state of the HMM set; though the proposed method required only 56% of the computational cost.

Future work will involve experiments using more accurate features and optimization of the combination of feature vector components to complement speech aspects.
Chapter 6

Conclusion and perspectives

6.1 Contributions and conclusions

This thesis described new dynamic approaches for the robust speech recognition. The conventional framework of speech recognition is based on a statistical paradigm. In a large vocabulary system, recognition hypotheses are scored by using two levels of statistical models: the language model that scores how likely a word sequence in a language, and the acoustic model that scores how likely a particular observation sequence in given a word sequence. Conventionally, the acoustic scores are calculated from a monolithic acoustic model, and integrated with the language likelihoods with a static weight. However, wide varieties of speech environments, speakers, and speaking styles, which comprise broadcasting contents, make the recognition difficult for practical application. This thesis proposed the methods of dynamic acoustic-scoring taking into account of such time-varying speech conditions.

Noteworthy results obtained through the study are as follows.

Chapter 1 made a review of the social circumstances surrounding the broadcasting service and the great demands for the speech recognition technology. It was brought out that the acoustical robustness against the varieties of the speech conditions is indispensable.

Chapter 2 reviewed the technical backgrounds of the large-vocabulary continuous speech
recognition. It was described with the mathematical representations that the recent approach on the probabilistic framework estimates the most probable linguistic representation of a given acoustic waveform. The proposed dynamic approaches were illustrated in association with this conventional framework.

Chapter 3 presented the dynamic selection of cluster dependent acoustic models. The method utilized a short and beginning fragment of an input utterance and GMMs corresponding the clusters to determine the most appropriate acoustic model. An efficient training algorithm was also proposed. The algorithm was based on the segment clustering in two stage, and it obtained specified sets of adaptation data and GMMs representing utterances in each corresponding cluster. It was concluded form the experimental results that the method not only reduced recognition errors but also reduced processing time while the method delayed starting the search process for the length of the beginning fragment. The method reduced the recognition errors especially in utterances of reporters in the evaluated news programs. This improvement was remarkable because such utterances are strongly demanded to improve their recognition accuracies so as to achieve wider use of this technology.

Chapter 4 presented the dynamic weights of the acoustic likelihoods. The acoustic likelihoods were weighted with their time-varying reliability which estimated from the amounts of the mismatch between training database and input speech. The mismatch was estimated by using the noise models incorporating knowledges of various kinds of noise. This chapter also proposed the method of generating the noise models automatically. Transcription experiments for broadcast sports commentaries showed that the method reduced word errors, especially in the keywords which are useful for the metadata extraction. The noise model generated by the method showed better result than those trained with hand-labeled noise clusters. It was concluded that the improvement given by the method should be of great advantage to the metadata extraction.

Chapter 5 presented dynamic integration of the likelihoods calculated from feature streams representing different acoustic aspects. This integration algorithm calculated a frame-wise stream weight so that a heavier weight was given to a stream that is robust to a variety of noisy environments or speaking styles. In order to weight such a robust stream that expected to bring out discriminative ability, the method utilized the entropy reduction caused
6.2 Perspectives

This thesis concluded that the dynamic approaches had the potential to improve recognition performance of broadcast contents. By integrating multiple statistic dynamically according to input speech, larger improvements were obtained from the proposed approaches than the conventional approach based on a monolithic statistics. These results seems to be reasonable because our brains achieve robust recognition without difficulty by shrewdly utilizes various experiences or knowledges consisting with a speech input. In the statistical speech recognition framework, the statistics play the role of representing these experiences or knowledges. On the other hand, the statistics are the joint probabilities of specific speech aspects in specific domains, that is to say, the combination of acoustic feature elements, linguistic elements, and domains of training data can be optimized so that the probability represents such experiences or knowledges. As shown in the results of this thesis, the recognition performance may be improved by the dynamic integration of such statistics representing the experiences.

This thesis individually presented the dynamic integrations of the elemental statistics taking into account of the heuristic knowledges such as the speakers, the corrupted speech by noise, or the acoustic feature streams. These knowledges, however, should be incorporated into a recognition framework and future work will involve the development of the framework of integrating the various experiences dynamically. Future work will also involve the optimization of the combination of statistical elements, such as feature vectors representing complement speech aspects, and it will be a great discovery in the speech recognition framework if one develops the method which efficiently comprise the statistics of representing our verbal experiences.
Bibliography


Publications

Major publications


Technical Reports


Oral Presentations


[8] Shoei Sato, Takeshi Kobayakawa, Kazuo Onoe, Shinich Homma, Akio Kobayashi,


