NAIST-IS-DD0561039

Doctoral Dissertation

Selective Training for Cost-Effective Development of Real-Environment Speech Recognition Applications

Tobias Cincarek

March 04, 2008

Department of Information Processing
Graduate School of Information Science
Nara Institute of Science and Technology

A Doctoral Dissertation
submitted to Graduate School of Information Science,
Nara Institute of Science and Technology
in partial fulfillment of the requirements for the degree of
Doctor of ENGINEERING

Tobias Cincarek

Thesis Committee:

Professor Kiyohiro Shikano	(Supervisor)
Professor Masatsugu Kidode	(Co-Supervisor)
Associate Professor Hiroshi Saruwatari	(Co-Supervisor)

Selective Training for Cost-Effective Development of Real-Environment Speech Recognition Applications[*]

Tobias Cincarek

Abstract

The most natural user interface for human-machine interaction is speech. Moreover, there are many applications for automatic speech recognition (ASR) technology, e.g. dictation systems, car navigation systems, real-environment guidance systems, dialogue robots, etc. ASR system have the difficulty that they are task- and domain-dependent. Consequently, the construction of an ASR system with reasonable performance usually requires large amounts of human-transcribed speech data collected in the target environment. However, collection and human labeling of large amounts of speech data is expensive and impractical whenever a new system for a new environment has to be built. Therefore, it is imperative to investigate more cost-effective development strategies. In literature several approaches to reduce costs of human-labeling such as unsupervised, lightly supervised and active learning have been proposed. Although these approaches have been effective in many cases, they do not address the aspect of task-dependency sufficiently. Therefore, one purpose of this work is to develop a cost-effective method for automatic construction of task-adapted acoustic models.

A framework for reuse of existing speech data and for selective training is proposed. To investigate the costs for developing a real-environment ASR application, a development simulation for the speech-oriented guidance system *Takemaru* installed at a community center is conducted first. The system's major components are an ASR module and a Q&A module. Since ASR task and Q&A domain are determined by the user and the system's environment, it is imperative to collect real-environment speech data. It is found empirically that about forty thousand utterances are required until performance saturates. In order to reduce the development costs of new systems for other environments, the effect of reusing the well-trained *Takemaru* prototype system in the *Kita* environment, a local subway station, is investigated. Experimental results show that the *Takemaru* ASR module is highly reusable for the *Kita* environment. On the other

[*] Doctoral Dissertation, Department of Information Processing, Graduate School of Information Science, Nara Institute of Science and Technology, NAIST-IS-DD0561039, March 04, 2008.

hand, the reusability of the Q&A module was rather low. However, Q&A performance improved remarkably after update with moderate amounts of *Kita* data. Moreover, from a comparison of from-scratch development and *Takemaru* reuse and update with *Kita* data it is shown that the development period can be reduced more than half and the development costs for data collection and human transcription more than 40%.

Furthermore, a selective training algorithm is developed. The algorithm makes it feasible to select a speech data subset similar to some task-specific data from a large pool of existing speech data automatically. Selective training is shown to be effective for constructing a preschool children acoustic model using school children speech and an elderly acoustic model using adult speech. A relative improvement in ASR performance of up to 10% over training without data selection is achieved. Furthermore, in order to reduce the development costs of acoustic modeling for a speech-oriented guidance system, the proposed approach is also applied to build adult and child-dependent models in case of an automatically transcribed speech data pool. The selective training algorithm effectively discards non-speech inputs, utterances with a wrong transcription and utterances from the wrong speaker group. Experimental results also show that it is possible to reduce development costs up to 40% without compromising ASR performance.

This work is an important contribution for more cost-effective ASR system development using existing speech data resources. The data requirements to construct the prototype of a real-environment ASR application and its adaptation to another environment have been investigated. Reusing the components of the existing prototype system helped to reduce development period and costs for the new environment. Furthermore, a computationally feasible selective training algorithm has been proposed and applied successfully to construct task-adapted acoustic models using only moderate amounts of task-specific data. The combination of selective training with unsupervised learning was also effective.

Keywords:

Speech Recognition, Speech Dialogue Systems, Real-Environment, Development Costs, Task-Dependency, Selective Training, Reusability, Adaptability

Acknowledgements

This dissertation is not only a result of my personal efforts but also the well-established working environment and the opportunities for discussion with laboratory members and advice from supervisors at the Acoustics and Speech Processing Laboratory, Graduate School of Information Science, Nara Institute of Science and Technology, Japan.

I would like to express my deep gratitude to Professor Kiyohiro Shikano, the thesis supervisor, for the opportunity to join the laboratory, for his valuable ideas, guidance and comments regarding my research work and the way of how to arrange academic presentations.

I would also like to express my gratitude to all other members of the thesis committee, Professor Masatsugu Kidode and Associate Professor Hiroshi Saruwatari for their valuable comments during public hearing and seminars.

I would especially like to express my deep gratitude to Assistant Professor Tomoki Toda for his continuous support, valuable advice regarding the research work and technical Japanese and the never-ending encouragements. Without his guidance, the outcome of my work would not have been as successful as it has been. I have been and will always be motivated to conduct research with him.

I would also like to thank Assistant Professor Hiromichi Kawanami for his beneficial comments especially regarding how to write technical Japanese.

I would like to thank all those members of Speech and Acoustics Laboratory providing the always working computer environment, assistance in case of technical problems and an atmosphere for constructive discussions making my life as researcher comfortable. I would especially like to express my appreciation to Dr. Akinobu Lee, who is currently Associate Professor at Nagoya Institute of Technology, Dr. Ryuichi Nisimura, who is currently Assistant Professor at Wakayama University, Dr. Shigeki Miyabe and Dr. Randy Gomez, who are Post-Doctorate Fellows at Nara Institute of Science and Technology, Mrs. Toshie Nobori, the secretary at Speech and Acoustics Laboratory, Mr. Goshu Nagino, who is a researcher at Asahi-Kasei Corporation, Mr. Yoshimitsu Mori, Mr. Yuu Takahashi, Mr. Yamato Otani and Mr. Keigo Nakamura, who are Ph.D. candidates at Nara Institute of Science and Technology.

I would also like to thank Dr. Satoshi Nakamura, who is president of ATR/NICT, Dr. Elmar Nöth, who is head of the speech group at Erlangen-Nuremberg University, Mr. Rainer Gruhn, who is a researcher at Harman/Becker, Dr. Tomoya Takatani, who is a researcher at Toyota Motor Corporation,

Mrs. Izumi Shindo, who is a researcher at NEC Corporation, Mr. Tomoyuki Kato who is a researcher at Asahi Kasei Corporation, Mr. Shota Takeuchi and Mr. Hiroyuki Sakai, who are currently enrolled in the master course at Nara Institute of Science and Technology and Dr. Jani Even, who is currently a Post-Doctorate Fellow at Acoustics and Speech Laboratory.

Finally, I would like to thank my relatives in Germany and Japan for their continuous support in everyday occurrences.

This work has been supported in part by the MEXT (Ministry of Education, Culture, Sports, Science and Technology, Japan) COE and e-Society projects.

Contents

1 Introduction **1**

1.1 ASR Technology . . . 2

1.2 ASR Applications . . . 3

1.3 Practical Challenges . . . 4

1.3.1 Task and Domain Dependency . . . 4

1.3.2 Development Costs of an ASR System . . . 5

1.4 Learning and Training Methods for ASR Systems . . . 7

1.4.1 Supervised Learning . . . 8

1.4.2 Active Learning . . . 9

1.4.3 Unsupervised Learning . . . 9

1.4.4 Lightly Supervised Learning . . . 11

1.4.5 Combinations . . . 11

1.5 System Portability . . . 11

1.5.1 Cross-Task Performance . . . 12

1.5.2 Multi-Source Training . . . 13

1.5.3 Task Adaptation . . . 14

1.6 Scope of This Work . . . 14

2 Fundamentals **17**

2.1 Statistics . . . 17

2.2 Learning Theory and Model Estimation . . . 17

2.2.1 Maximum Likelihood Estimation . . . 18

2.2.2 Sufficient Statistics . . . 19

2.2.3 Expectation-Maximization Algorithm . . . 19

2.3 Hidden Markov Model . . . 21

2.4 Automatic Speech Recognition . . . 24

2.4.1 Speech and Transcription . . . 24

2.4.2 Feature Extraction . . . 26

2.4.3 HMM-based Acoustic Model . . . 27

2.4.4 Statistical Language Model . . . 29

2.4.5 Decoder . . . 30

2.5 Evaluation Measures . . . 31

2.5.1 OOV Rate and Perplexity . . . 31

2.5.2 Recognition Performance . . . 31

2.5.3 Response Accuracy 32
2.5.4 Development Costs 32
2.6 Text Matching 32

3 Data, Model and System Reuse 35
3.1 Spoken Dialogue Systems 36
3.2 Speech-Oriented Guidance Systems 38
3.2.1 Purpose and Architecture 38
3.2.2 Takemaru 39
3.2.3 Kita-chan and Kita-robo 40
3.2.4 Real-Environment Speech Database 41
3.3 Development of Prototype System 41
3.3.1 Experimental Setup 42
3.3.2 Performance Evaluation 44
3.3.3 Conclusion 48
3.4 Reuse of Prototype System 49
3.4.1 Prototype System 50
3.4.2 Task Data for System Update 51
3.4.3 Experimental Results for System Update 52
3.4.4 System Update vs. From-Scratch Development 53
3.4.5 Takemaru Development vs. Kita System Update 54
3.5 Domain Analysis and Comparison 55
3.5.1 Domain Distance 56
3.5.2 User Utterance Contents 58
3.5.3 Related Work 59
3.6 Conclusion 61

4 Selective Training for Task-Adaptation 63
4.1 Conventional Data Selection Methods 63
4.2 Proposed Data Selection Method 64
4.2.1 Selective Training Framework 64
4.2.2 Optimization and Selection Criterion 65
4.2.3 Selection Algorithm 70
4.2.4 Graphical Illustration of Selective Training 72
4.2.5 Remarks and Conclusion 76
4.3 Possible Applications 77

5 Experiments with Selective Training 79
5.1 Human-Transcribed Data Pool 79
5.1.1 Comparison of Algorithm Variants 80
5.1.2 Comparison of Acoustic Models 84
5.1.3 Conclusion 88
5.2 Untranscribed Data Pool 89
5.2.1 Target Application 90

5.2.2 Problem Description 91
5.2.3 Proposed Training Procedure 92
5.2.4 Simulation Experiments 93
5.2.5 Experimental Results 96
5.2.6 Conclusion 99

6 Summary 101
6.1 Data, Model and System Reuse 101
6.2 Selective Training for Task Adaptation 102
6.3 Selective Unsupervised AM Training 103
6.4 Outlook and Future Work 103

A Experimental Results 105
A.1 Takemaru Development Simulation 105
A.2 Kita-chan and Kita-chan Robot Adaptation 106
A.3 Selective Unsupervised Training 107

B Glossary 109

References 111

Publications 119

List of Figures

1.1 Evolution of user interfaces. 2
1.2 Examples of applications for ASR technology. 4
1.3 Basic architecture of an automatic speech recognition system . . . 6
1.4 Problems when developing ASR applications. 6
1.5 Illustration of conventional learning methods. 10
1.6 Classification of Current Work. 15

2.1 Illustration of left-to-right HMM with output densities 21
2.2 Speech signal and MFCC feature vector sequence 26
2.3 Conversion of speech signal from time-domain to frequency domain. 27
2.4 Conversion of short-term spectral coefficients to MFCC 27
2.5 Example for acoustic model with state-tied triphone HMMs. . . . 28
2.6 Structure of acoustic model with phonetically tied mixtures. . . . 29

3.1 Task and Domain-Dependency of Real-Environment Applications. 35
3.2 Task and Domain-Dependency of Real-Environment Applications. 36
3.3 Main building blocks of a speech-oriented guidance system. 37
3.4 Speech-oriented guidance system *Takemaru*. 38
3.5 Inputs collected with *Takemaru* during two years of operation. . . 39
3.6 Speech-oriented guidance systems *Kita-robo* and *Kita-chan*. 40
3.7 Monthly input statistic for *Kita-chan* and *Kita-robo* 41
3.8 Number of training utterances and distinct words 43
3.9 OOV rate and LM perplexity for *Takemaru* 45
3.10 Effect of updating the *Takemaru* LM 45
3.11 Effect of updating the *Takemaru* AM 46
3.12 Number of distinct example queries and response sentences. . . . 47
3.13 *Takemaru* Response Accuracy 47
3.14 System reuse and portability 49
3.15 Performance comparison for different development strategies . . . 53
3.16 Result of portability experiment for *Takemaru* → *Kita* 55
3.17 Cross-domain histogram of relative response frequencies. 57
3.18 Subdomain analysis and comparison. 58

4.1 Proposed selective training framework. 65
4.2 Likelihood-based selection condition. 67

4.3 Setup and preparations for selective training 70
4.4 Illustration of the steps of the selective training procedure 71
4.5 Conditions for simulating the behavior of selective training. 73
4.6 Behavior of selective training after few and many iterations 74
4.7 Comparison conventional EM training vs. selective training 76
4.8 Task-adapted acoustic modeling using selective training 78
4.9 Improving unsupervised training with selective training 78
4.10 Cross-language acoustic modeling using selective training 78

5.1 Likelihood distribution given selected vs. discarded utterances . . 82
5.2 Behavior of the Q-function (Experiment B, *ST_DelAdd*). 83
5.3 Comparison of selective training and adaptation 84
5.4 Influence of the amount of task-specific speech data 86
5.5 Summary of experimental results 89
5.6 Speech-oriented guidance system *Takemaru* 90
5.7 Inputs collected during the first two years of operation. 91
5.8 Selective training using greedy ML data selection strategy 92
5.9 Framework for cost-effective acoustic model construction 93
5.10 Procedure for acoustic model construction. 94
5.11 Performance for adult and child AM (one week transcription). . . 97
5.12 Performance for adult and child AM (18 month data collection). . 97

A.1 Evaluation of unsupervised selective training (adults) 108
A.2 Evaluation of unsupervised selective training (children) 108

List of Tables

1.1 Conventional learning methods 7
1.2 Training algorithms for acoustic modeling 8
1.3 ASR performance with broadcast news vs. task-specific models . . 12
1.4 Experimental results for cross-task performance evaluation 13
1.5 ASR performance with multi-source acoustic and language models 13
1.6 Supervised and unsupervised adaptation of acoustic models 14

2.1 Phoneme set of the Japanese language 25
2.2 Mapping of selected Katakana to phonemes 25
2.3 Examples of Japanese graphemes 25
2.4 Examples for Japanese morpheme sequences 26
2.5 Assumed system development base costs and cost factors 32

3.1 Speech data collected with *Takemaru* and *Kita* 42
3.2 Data employed for developing the *Takemaru* ASR module 42
3.3 Experimental conditions for development simulation 43
3.4 Question and answer database of *Takemaru* 44
3.5 Response accuracy for *Takemaru* subdomains 48
3.6 ASR and Q&A performance of *Takemaru* 50
3.7 Evaluation of *Takemaru* LM 51
3.8 Training and evaluation data collected in *Kita* environment 51
3.9 Contents of *Kita* QADB 52
3.10 ASR and Q&A performance of *Kita* 52
3.11 Evaluation of *Kita* LM 52
3.12 Cost performance of *Kita* systems 54
3.13 Domain comparison using response statistics 57
3.14 Domain comparison using language models 57
3.15 Response accuracy for *Kita* subdomains 59
3.16 Responses very common in *Takemaru* and *Kita* domain 60
3.17 Responses relatively frequent for *Takemaru* domain 60
3.18 Responses relatively frequent for *Kita* domain 60
3.19 Keywords in user utterances 61

4.1 Updated model parameters after each selective training iteration . 75
4.2 Q-function's value before and after each selective training iteration 76

5.1 Data pool and task-specific development data for experiments . . 79
5.2 Conditions for evaluating the two variants of selective training . . 80
5.3 Comparison of conventional vs. selective training for preschool data 81
5.4 Comparison of conventional vs. selective training for elderly speech 81
5.5 Number of utterances selected by proposed algorithm 81
5.6 Relationship between development set size and performance . . . 82
5.7 Space and time requirements of selective training 84
5.8 Comparison of Acoustic Model Parameters 85
5.9 Experimental Results for Monophone and PTM Acoustic Model . 85
5.10 Relationship between amount of task-specific and selected data . . 87
5.11 Performance of High-Cost Models 87
5.12 Summary of Experimental Results 88
5.13 Experimental conditions for selective, unsupervised training. . . . 95
5.14 Human-labeled part of the *Takemaru* database 95
5.15 Labeled and unlabeled training and evaluation data 95
5.16 Data pool statistics and selection results 98
5.17 Breakdown of selected training data 98
5.18 Comparison of supervised learning and proposed method 99

A.1 Results for Takemaru Development Simulation (Adults) 105
A.2 Results for Takemaru Development Simulation (Child Data) . . . 106
A.3 Results for Kita-chan Adaptation (Adult Data) 107
A.4 Results for Kita-chan Adaptation (Child Data) 107

Chapter 1

Introduction

With the emergence of electro-technical devices (machines) came the necessity to provide humans an interface for interaction. For example, buttons of radios, TV sets, coffee-makers, washing machines, etc. can be considered as the earliest human-machine interfaces. Since machines evolved over time and applications for various purposes were developed, there was the necessity to built interfaces which fit the needs of potential users. The invention of digital computers and its miniaturization is the invention of the 20th century with the greatest influence on human life today. People send e-mails with their cellular phone or PDA instead of writing letters with pen and paper, check Internet webpages for the latest news using their personal computer instead of reading the daily newspaper, download a song of their favorite artist from an online provider instead of buying a record at a music store, use a GPS-based car navigation system instead of looking at a paper road map and so on.

In the beginning, however, home computers and hand-held devices were not very powerful and convenient. In the first operating systems there were only command line interfaces (CLI) with line-wise input and output. Only a keyboard served as input device by that time. In the next development step, text(ual) or terminal user interfaces (TUI) came up which used the entire screen to simulate a higher-level interface with special text symbols as precursor to graphical user interfaces (GUI). Later, today's GUIs emerged which provide a window-based workspace. The computer mouse was also employed as additional input device. Besides the mouse, graphics tablet for painting, touch-pad and touch-screen have become accepted widely among users.

Despite the tremendous development and progress made, there are situations in which the above-mentioned array of user interfaces is not sufficient yet. Imagine you would like to change the current destination of the car navigation system while driving, write an e-mail while you are taking a bath, change the TV channel or currently played song although your hands are syrupy while you are preparing a meal or the washing dishes, or as a mother you would like to switch off the gas burner or control the temperature settings of your air-conditioner while holding and feeding your newborn child in your arms. In these situations it is

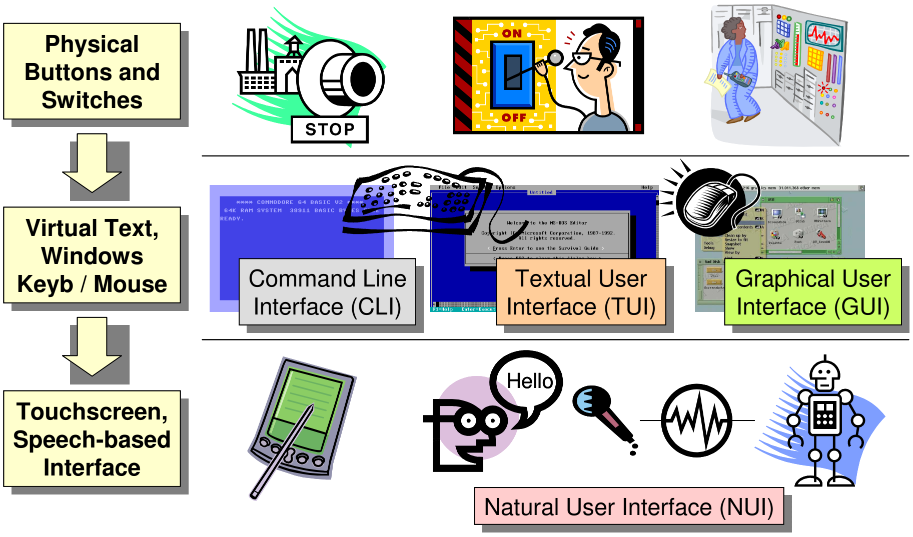
Physical Buttons and Switches
STOP
ON
OFF
Virtual Text, Windows Keyb / Mouse
Command Line Interface (CLI)
Textual User Interface (TUI)
Graphical User Interface (GUI)
Touchscreen, Speech-based Interface
Hello
Natural User Interface (NUI)

Figure 1.1. Evolution of user interfaces.

difficult to use your hands or even approach the device which you like to control. Consequently, a more universal user interface is needed.

It would be possible to overcome these difficulties with a speech-based, natural user interface (NUI). Furthermore, speech is the most natural and efficient way for humans to communicate with each other. Everyday's life would become much more convenient if the interactions between humans and machines were based on speech. To realize a speech-based user interface two key technologies are required: speech recognition to convert a recorded speech signal into text and speech synthesis to generate a speech signal from text. Speech synthesis technology can already be considered as mature to be employed for developing practical applications. However, although research has been conducted for more than 50 years there are still practical challenges for speech recognition technology.

1.1 ASR Technology

We start with a brief excerpt from [24] of the history of automatic speech recognition. Research on speech recognition started in the U.S. and later spread to many other countries. A device to recognize digits was developed in 1952. Work by different researchers and institutions for recognizing vowels, syllables and a small number of words followed. Recognition was first based on the detection of resonant frequencies of the human vocal tract (formants). Later a detailed spectral analysis of the speech signal became more important.

In the 1960s and beginning of the 1970s spectral estimation techniques such as fast Fourier transform (FFT) and linear predictive coding (LPC) were developed. For recognition, a pattern matching approach based on dynamic time-warping (DTW) was employed to cope with the varying length of utterances. A sequence of spectral estimates is compared with a template constructed from several utterance examples of a word. Initial speech recognition systems were most often speaker-dependent, i.e. only speech from a speaker the system has been trained on could be recognized with a reasonable performance.

In the late 1970s and 1980s, researchers turned towards statistical approaches using hidden Markov models (HMM) for acoustic modeling and N-grams for language modeling. The goal of a government-funded research project was to built a speech recognizer with a vocabulary of 1,000 words. In order to provide the research community a common database for research and system constructed, standard speech corpora have been collected from the 1980s. With large speech databases it was possible to scale the existing technology. Major English corpora are TIMIT (phonetically balanced sentences), Resource Management (commands and questions) and WSJ (read articles from Wall Street Journal). The vocabulary size for recognition increased from 1,000 to 5,000 and 20,000 words. With a large number of utterances from many speakers in the database it was also possible to build speaker-independent systems.

In the 1990s the focus shifted to recognition of broadcast news, conversational speech and lecture speech. The major difference to previous speech recognition tasks is the change from read to spontaneous speaking style which makes speech recognition much more difficult due to more acoustic variability. The recognition vocabulary size was increased further to 60,000 words. The difficulty with speaker-independent systems is that their performance is inferior to speaker-dependent systems. From this arose the necessity to develop methods for speaker adaptation.

In the beginning, speech recognition experiments were only performed in laboratory conditions, i.e. the speech data has been very clean with almost no background noise. In real-environment and hands-free conditions, however, there is much interference from background noise, background conversation, reverberation, etc. Therefore, denoising and dereverberation techniques have also been developed to realize noise-robust speech recognition systems.

Recently, researchers are considering speech recognition for various tasks, e.g. different languages [67], non-native speech [81, 13], dialected speech [17], children speech [60], preschool children speech [14], non-audible speech [29] and so on. After 2000, real-time systems with a recognition vocabulary of more than one million words have been realized by using weighted finite-state transducers (WFST) [30].

1.2 ASR Applications

The application of ASR technology is not restricted to human-computer interfaces. Over the years, many systems and applications have been devised and developed. Since in the beginning ASR systems could only handle a limited number of words, only simple applications like voice dialing or command recogni-

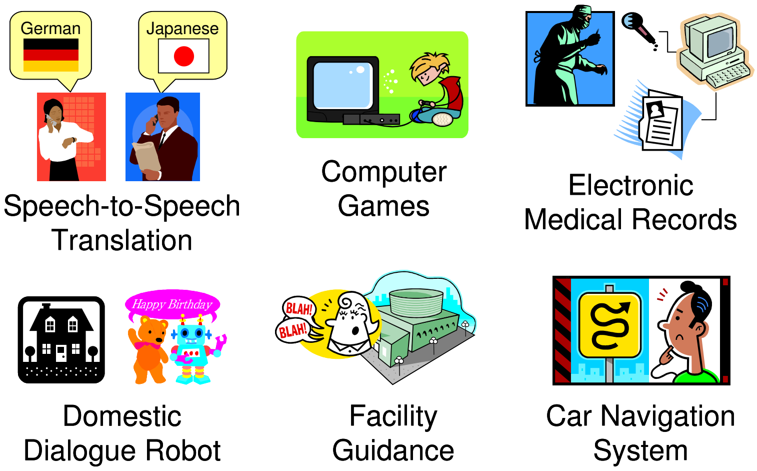

Figure 1.2. Examples of applications for ASR technology.

tion for controlling devices were possible. As vocabulary size increased, dictation systems could be built. At IBM [33] a voice-activated typewriter Tangora [15] has been developed. Today, Nuance [57] sells IBMs ViaVoice [78] dictation system. Moreover, Nuance and Advanced Media [2] sell dictation systems for transcriptions of medical records. More possible domains for automatic transcription are broadcast news, plenary sessions, lectures and meetings. SpinVox [71] sells a system for converting voice messages into text.

Further applications are goal-oriented dialogue systems like the air travel information system (ATIS) for airline ticket reservation [85], speech-to-speech translation systems for travel expressions [73], speech-oriented guidance systems [55], automation of call centers, interface for interactive computer games, dialogue robots and CALL systems for foreign language pronunciation training (Figure 1.2).

1.3 Practical Challenges

Speech recognition is task- and domain-dependent. Therefore, it is difficult to achieve a high recognition performance for different tasks with a single speech recognition system. Consequently, data collection, data preparation and model training for the target task and domain are important in practice. However, the development costs of a task-adapted ASR system are high due to data collection and preparation. Therefore, the construction of portable, i.e. reusable and adaptive systems is a major concern. In the following these issues will be discussed more deeply.

1.3.1 Task and Domain Dependency

There are practical challenges for developing an ASR application which is mainly due to the task and domain-dependency of speech recognition. Speech recognition performance depends on various factors:

1. speaker characteristics (age, gender, accent, etc.)
2. speaking style (read, spontaneous)
3. opponent (human, machine)
4. transmission channel (telephone, microphone, hands-free)
5. target domain (commands, dictation, dialogue)

The acoustic characteristics of speech depend on a speaker's age and gender due to differences in human vocal tract length and frequency of excitation pulses from the glottis. While in read speech phonemes are clearly pronounced, coarticulation effects are strong in spontaneous speech with deletions and deformation of speech sounds. The characteristics of speech also depend on whether we speak to humans, e.g. lecture speech, conversational speech, or a machine, e.g. dialogue system speech.

Furthermore, the quality of speech depends on the transmission channel. Reverberation and noise from the target environment, the recording device, microphone and other analogue and/or digital processing may have a negative influence on speech quality. To enhance the speech quality before actual speech recognition, signal processing techniques for dereverberation and denoising have been developed as frontend, e.g. spectral subtraction [8, 7] and microphone array systems [19].

Finally, there is the application domain with its main influence on the utterance contents, but less on the acoustic characteristics of speech.

Considering all these factors, it is obviously impossible to build a universal speech recognition system with a high recognition performance independent from the target application. Therefore, it is necessary to customize the speech recognizer for each application. However, the costs for customization are high as discussed in the following.

1.3.2 Development Costs of an ASR System

The basic architecture of a speech recognition system is shown in Figure 1.3. Input speech is recorded via a microphone in a real environment. In case of a noisy environment or long reverberation time, signal processing for denoising and dereverberation may have to be applied. Features related to the human vocal tract, which are important for automatic recognition, are extracted from the eventually denoised and dereverberated speech signal. The decoder employs a search algorithm to find the most likely spoken word sequence $\mathbf{W}$ given the acoustic observation $\mathbf{X}$.

The acoustic model determines the acoustic characteristics of the speech to be recognized, the language model the words and structure of sentences to be

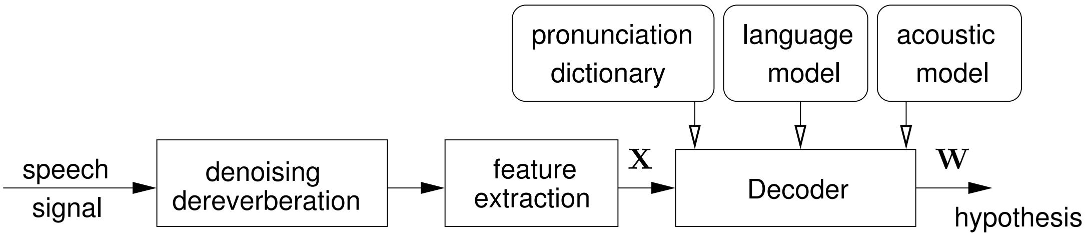

Figure 1.3. Basic architecture of an automatic speech recognition system.

recognized. The pronunciation dictionary provides the mapping from tokens of the language model (words) to units of the acoustic model (phonemes). Standard speech recognition systems today employ statistical acoustic and language models with a large number of parameters. To estimate the model parameters reliably, a huge amount of training data is necessary.

The preparation of large amounts of training speech data is most often very costly, because they have either to be collected in a real environment or subjects have to be called for speech recordings. Furthermore, the collected data has to be segmented, transcribed and labeled by humans in order to achieve good model training results (Figure 1.4).

Although collection of the speech data in a real-environment requires a working prototype system, it is relatively easy to obtain various kinds of inputs from many potential users. The drawback is that humans have to listen to each recorded input and transcribe it manually. In case of calling subjects for speech recordings, efforts for speech data transcription can be reduced to a minimum because the sentences each subject should read can be prepared in advance. However, calling subjects itself is expensive and it is difficult to obtain a large diversity of utterances.

For example, an analysis from IBM [21] shows that about about half of the costs of building an interactive dialogue system are due to the speech recognition component. Data collection, transcription and annotation account for about 40% of the costs. An investigation of Asahi-Kasei [52] amounts the costs for calling a single a subjects for speech recordings to about \$400. From a personal experience of the author as intern at ATR, Spoken Language Translation Research Labs, the costs for calling a subject for two hours where about \$250. Furthermore, scientific publications state that the transcription of speech takes 20-40 times real-time, i.e. duration of the recorded speech data [42, 83].

1.4 Learning and Training Methods for ASR Systems

Although the technical terms 'learning' and 'training' are often used in the same sense, their meaning is distinguished in the following. 'Learning' refers to the manner of learning, that is whether and how the machine learning process is

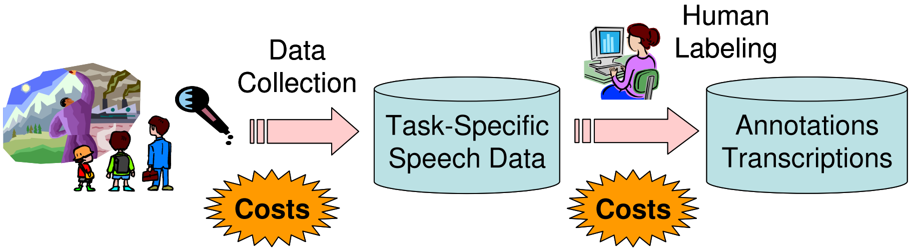

Figure 1.4. Problems when developing ASR applications.

supervised by humans. 'Training' refers more to algorithmic aspects or to the sources of the training data.

The development of an ASR system for a speech-oriented application requires the construction of the acoustic as well as the language model. They are equally important for system development with respect to recognition performance. However, many previous research activities on cost-reduction of ASR development have focused mainly on acoustic modeling taking either the existence of an appropriate language model for granted or it was relatively easy to obtain a domain-specific language, e.g. broadcast news [42, 80, 40, 49]. Moreover, it is inherently more difficult to obtain labeled training for acoustic modeling, because both collection of speech data and its transcription are usually involved. Although depending on the task, there are many cases in which training data for the language model can obtained from webpages, newspapers, documents, etc.

An overview of learning methods for acoustic modeling is given in Table 1.1 and Figure 1.5. The learning methods differ with respect to the initial state of the available data (labeled, unlabeled), whether transcription of unlabeled data is conducted by humans or automatically and the selection criterion (none, confidence-based, agreement-based) for the initially unlabeled data.

Besides the learning method a further aspects of model construction is the training method. There are many aspects of training methods such as the algorithm for parameter estimation and the overall setup and procedure.

An overview to the most well-known and widely used algorithms for parameter estimation of HMM-based acoustic models is given in Table 1.2. They are

Table 1.1. Conventional learning methods, especially for constructing the acoustic model of an ASR system

Learning	Initial Data State		Selection	Utterance
Method	Labeled	Unlabeled	Criterion	Transcription
Supervised	○	×	none	manual
Active	×	○	confidence	manual
Unsupervised	×	○	confidence	automatic
Lightly Supervised	△	△	agreement	automatic
Semi-Supervised	○	○	confidence	partial

Table 1.2. Training algorithms for acoustic modeling

Training Algorithm	Optimization Criterion	Algorithm Category	Data Amount for Training
Baum-Welch [6]	ML	E.-M.[16]	Large
MAP [22]	MAP	E.-M.	Medium
MLLR [48]	ML	E.-M.	Small
Discriminative [56]	MMI	Gradient Descent	Large

implemented in the latest version of the Hidden Markov Model ToolKit (HTK) [31]. If large amounts of training data are available, the Baum-Welch algorithm [6] is usually employed for separate estimation of all HMM parameters. If only moderate amounts of data are available, it is better to use MAP or MLLR estimation to avoid overtraining, i.e. too small co-variances due to data insufficiency. In case of MAP estimation [22], a prior probability is assigned to the current model parameters. MLLR [48] requires to determines few parameters of a transformation (shift, shear and rotation) for mean vectors and co-variance matrices. It can even be carried out with very small amounts of data. There are also discriminative training methods which promise a better estimation of model parameters using the maximum mutual information (MMI) criterion [56].

In case of unsupervised learning there are self-training and co-training as training procedures. In self-training there is only one model or classifier to determine labels for the unlabeled data. For acoustic model training this means, that an initial acoustic model is employed to transcribe the unlabeled data. The initial model may eventually be adapted or trained with the available labeled data. The automatically labeled data are then used to retrain the initial acoustic model.

In co-training there are two classifiers to determine labels for the unlabeled data. The classifiers may either be trained on different portions of a labeled data set or trained using different feature sets. Or there may be labels for the unlabeled data from two different sources. Finally, only those unlabeled data for which both classifiers or both sources of information agree on the same label are employed for retraining. There is also agreement learning which employs only those unlabeled data for which multiple classifiers agree on the same label.

In the following, a survey on several learning methods using self-training for acoustic modeling is given. Sometimes different authors employ different names for the same learning paradigm. To clarify the matter, we shall stick to the classification scheme of learning methods given in Table 1.1 throughout this work.

1.4.1 Supervised Learning

In case of supervised learning all available data collected for a certain task domain are labeled by humans. It can guarantee a good performance if the human annotators are reliable. However, supervised learning with large amounts of data is often infeasible due to the costs of the human effort. It is also not enough to collect and prepare a large speech database once, because there is the task and domain-dependency of speech recognition which has already been outlined in Section 1.3.1. Alternative methods to supervised learning have to be considered in practice. Conventional methods and development strategies are described in the following.

1.4.2 Active Learning

The idea of active learning is to not label all collected data by humans from the beginning, but only to label (transcribe) the subset which is difficult to classify (recognize) with the existing classifier (acoustic model). The procedure of active learning for a speech recognizer is as follows: If there is no initial ASR system available which could be employed for the intended task and domain, transcribe a small set of the unlabeled data to bootstrap an initial acoustic and/or language model. Next, transcribe all available unlabeled data automatically with the initial ASR system. Only a subset of the unlabeled utterances with a low recognition confidence are finally selected for human transcription. After human transcription of the selected data, the initial models are retrained or reconstructed with the extended labeled data set. This procedure can be iterated several times.

Experiments with active learning showed that the same performance can be reached with human transcription of about half to one third of the unlabeled data in comparison to when transcribing all data by humans [38]. It was even possible to improve the performance over when transcribing and using all data for model training. Further examples and experiments of applying active learning for speech recognition and understanding are given in [28, 77].

1.4.3 Unsupervised Learning

The purpose of unsupervised learning is to avoid human transcription of collected speech data at all. This is obviously the most effective way to reduce development costs. However, there is also the drawback, that it is difficult to obtain the same performance as with supervised learning. Unsupervised learning requires automatic transcription of the unlabeled data with an existing ASR system. Since speech recognition errors are inevitable, automatic transcriptions will always be error-prone. Therefore, it is worth considering to employ only a certain subset of the unlabeled data for model training.

In contrast to active learning, where utterances with a low recognition confidence are selected for human transcription, correctly transcribed utterances are more important for unsupervised learning. This means that utterances with a high recognition confidence should be selected.

Experiments with unsupervised learning have been carried out, e.g. for transcription of broadcast news [80, 40] or for porting a broadcast news transcription system to conversational speech [23]. Only very small amounts of transcribed speech data were employed for bootstrapping initial models.

Unfortunately, in several investigations of unsupervised learning the performance is only shown for a few selected amounts of unlabeled data and comparison is only carried out for a few selected amounts of labeled data [42, 49]. Moreover, there are investigations where the number of acoustic model parameters is increased depending on the amount of training data. Therefore, it cannot be assessed clearly whether performance is due to a larger number of parameters or adding the unlabeled data for retraining [80, 49]. These circumstances make it difficult to draw definite conclusions about the effectiveness of unsupervised learning.

Although unsupervised learning seems to work, there are no clear empirical results for the selection of utterances with high quality transcriptions using confidence measures. The higher the confidence value, the higher the probability that the automatic transcription is correct. However, findings from different investigations are controversial. There are also results which show that it is better to select utterances with medium or high confidence [40, 80] as well as results which suggest the selection of utterances with a low recognition confidence [83].

From other investigations it is not even clear whether confidence-based selection significantly improves unsupervised learning [49]. The underlying reason is that confidence-based selection tends to incorporate data which is already represented well by the model and that the estimation process does not converge to the true model parameters [84]. This is due to the fact that the confidence measures are often calculated as the posterior probability using an existing classifier.

The performance gap between unsupervised and supervised learning is still relatively large, even if large amounts of unlabeled speech data are available. Moreover, it seems to be easy to outperform unsupervised learning with moderate amounts of human transcribed data [79].

1.4.4 Lightly Supervised Learning

Lightly supervised learning combines the cost advantages of unsupervised learning with existing knowledge sources. Since the provision of accurate transcripts for speech data is expensive, they should be avoided as much as possible. On the other hand, performance could be improved remarkably, if ground truth would be available.

In case of television shows and broadcast news, less expensive closed-captions are often available. The closed-captions are approximate transcriptions of what is being spoken. Although they are mainly provided for speech-impaired people, they are also very helpful for non-native listeners. As additional knowledge sources, content-related texts can be employed, e.g. newspaper texts and scripts.

The initial acoustic model of the speech recognizer is bootstrapped with a small amount of human transcribed speech data. Furthermore, a domain-specific language model is constructed from all existing closed-captions and content-

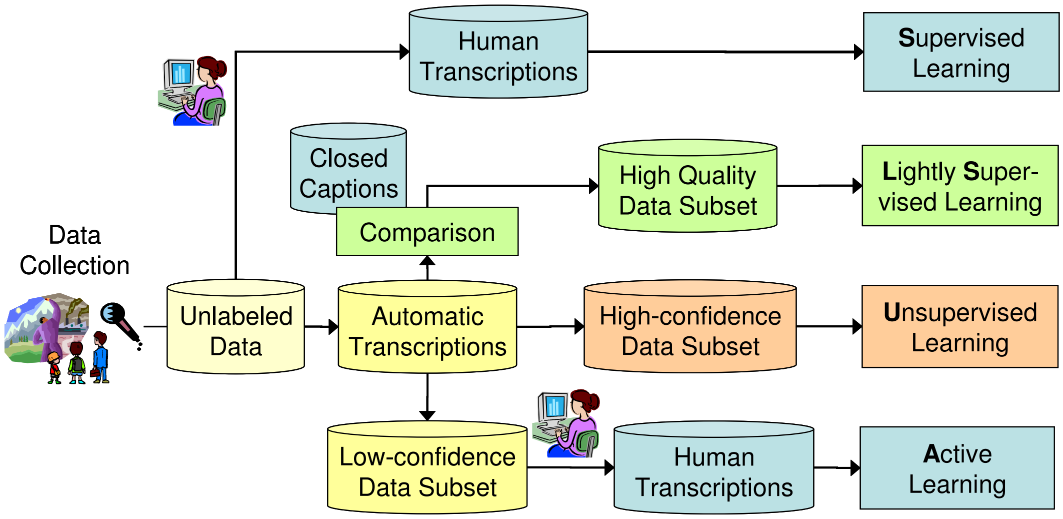

Figure 1.5. Illustration of conventional learning methods.

related texts. Using these models, all unlabeled speech data available are automatically transcribed as for unsupervised learning. But instead of considering all utterances or utterances with a high confidence, only speech segments matching the closed-captions are employed for training [41].

The requirement of an exact agreement between closed-captions and recognition hypothesis may result in training data loss and it is likely that mainly utterances which can already be recognized well by the system are employed for retraining. However, this would reduce the learning effect from unlabeled data. Therefore, an improvement of lightly supervised learning has been proposed [11]. The requirement of full agreement between the automatic transcript (first best hypothesis) and closed-caption is relaxed. Either the posterior probability of words in the consensus hypothesis [50] is higher than a certain threshold, or the closed-caption is part of the recognition lattice. Experimental results showed an improvement over conventional lightly supervised training.

1.4.5 Combinations

It is possible to combine learning methods. For example, it is possible to combine active and unsupervised learning. The utterances selected for human labeling are employed for supervised learning and the remaining utterances for unsupervised learning. It has been shown experimentally, that this combination is better than active learning alone until approximately three thousand human-transcribed utterances have been employed for training [65].

Table 1.3. Speech recognition performance (Word error rate) when using broadcast news (BN) models versus task-specific models. Adapted from [43]

Evaluation Task Name	Description of Recognition Task	BN AM BN LM	BN AM Task LM	Task AM Task LM
BN	TV+Radio News	13.6%		
TI-Digits	Connected Digits	17.5%	1.7%	0.4%
ATIS	Human-Machine Dialog	22.7%	4.7%	4.1%
WSJ Read	News Dictation	11.6%	9.0%	7.6%
WSJ Spon	News Dictation	12.1%	13.6%	15.3%

1.5 System Portability

The discussion of learning and training methods in the last section was largely based on the premise that the construction of an ASR application often starts from scratch. However, it is often possible to reuse existing speech corpora, models and ASR systems. Furthermore, it is worth to aim at building systems which are portable among different tasks and domains. Portability means reusability as well as adaptability.

Therefore, a report on task- and domain-dependent differences in recognition performance is given in the following. Moreover, results from literature for the construction of a task-independent system using multi-source training are described. Finally, experimental results for supervised and unsupervised task adaptation of acoustic models are cited.

1.5.1 Cross-Task Performance

In Section 1.3.1 the reasons for the task- and domain-dependency of speech recognition have been mentioned. In the following the degree of dependency is investigated in terms of difference in recognition performance between matched and mismatched model conditions. Excerpts from literature as well as results of experiments for cross-task evaluation are reported.

Table 1.3 shows the recognition performance for five different tasks using broadcast news (BN) acoustic model (AM) and language model (LM) versus task-specific AM and LM. It is clear that the dependency on the domain, i.e. language model, is higher than the dependency on the acoustic model. That the accuracy for spontaneous news dictation is higher with BN models than with task-specific models is not surprising since there are obvious similarities between spontaneous news dictation and TV or radio news.

In order to investigate task-dependent performance difference due to speaking style, a cross-task performance evaluation for Japanese newspaper dictation and human-machine dialogue has been conducted. The former is a read speech, the latter are spontaneous speech recognition tasks. Although the acoustic environments differ, the influence can be considered as small since either a close talking

Table 1.4. Experimental results for cross-task performance evaluation in case of different speaking styles and speaker characteristics (Word accuracy)

Evaluation Task	JNAS Model	Takemaru Model	Kitachan Model
JNAS (Dictation)	91.2%	77.8%	73.9%
Take (Human-Machine Dialogue)	64.6%	70.1%	70.0%
Kita (Human-Machine Dialogue)	73.6%	79.0%	79.7%

Evaluation Task	Adult Model	Hybrid Model	Child Model
Adult Speech	70.1%	65.4%	55.9%
Child Speech	38.0%	53.8%	54.6%

or directivity microphone have been employed for recording. Furthermore, only adult speech data has been employed for the evaluation to exclude influence due to speaker characteristics.

In the evaluation only performance differences due to the acoustic model are investigated. The language model has always been selected to be domain-specific. The result in Table 1.4 shows that it is difficult to recognize read speech with a spontaneous speech AM and vice versa.

1.5.2 Multi-Source Training

There have also been quite successful attempts to construct task-independent acoustic and domain-independent language models. Instead of optimizing the ASR system for one task, constructing a universal ASR system by combining several speech and text databases from multiple sources has also been investigated [47]. A performance comparison between task-specific and multi-source models is given in Table 1.5.

There is an improvement for spontaneous news dictation using the multi-source AM and LM. Moreover, the performance increases for read news dictation and human-machine dialogue using a multi-source acoustic model. However, thee is a degradation in accuracy for digit recognition and broadcast news transcription. It can be concluded that multi-source acoustic models can work quite well for several tasks except digit recognition. The employment of multi-source language models was only promising for recognition of spontaneous news dictation and human-machine dialogue.

1.5.3 Task Adaptation

The effect task adaptation of a broadcast news acoustic model with task-specific data using supervised and unsupervised learning as investigated by [47] is shown in Table 1.6. Language models are selected to be task-specific. Unsupervised

Table 1.5. Comparison of ASR performance (Word error rate) between task-specific and multi-source acoustic and language models. Adopted from [47]

Evaluation Task	BN	TI-Digits	ATIS	WSJ Read	WSJ Spon
Task-specific AM + LM	13.6%	0.4%	4.1%	7.6%	15.3%
Multi-Source AM, Task LM	14.9%	0.7%	3.1%	6.7%	11.8%
Multi-Source AM + LM	17.5%	-	4.0%	8.6%	11.2%

Table 1.6. Supervised and unsupervised adaptation of broadcast news acoustic models with task-specific data. Evaluation is conducted with task-specific language models (Word accuracy). From [47]

Acoustic Model	TI-Digits	ATIS	WSJ Read	WSJ Spon
Broadcast News Models	1.7%	4.7%	9.0%	13.6%
Unsupervised Adaptation	0.8%	4.7%	6.9%	11.9%
Supervised Adaptation	0.5%	3.2%	6.5%	11.0%

adaptation seems to be effective for all tasks but human-machine dialogues (ATIS). Supervised adaptation is effective for all tasks. The difference between supervised and unsupervised adaptation are relatively small except for human-machine dialogues. Furthermore, the effectiveness of unsupervised adaptation depends heavily on the language model for automatic transcription. However, for many tasks it is difficult to obtain a high-quality LM to determine accurate transcriptions.

1.6 Scope of This Work

Speech recognition technology is of interest to realize more natural human-machine interfaces. An overview to automatic speech recognition (ASR) technology and its applications has been given in Sections 1.1 and 1.2. Furthermore, it has been outlined in Section 1.3 that one of the reasons of the currently limited success of products with ASR technology are the task and domain dependency and the high development costs for high-performance, real-time ASR applications.

The research community has made efforts to reduce development costs and made attempts to build systems with increased portability. Conventional learning and training methods for ASR systems have been described in Section 1.4. There are lightly supervised, active and unsupervised learning. These conventional methods for more cost-effective system development have the drawback that they do not consider the aspect of task and domain-dependency sufficiently. Nor do they provide means to directly construct a task-adapted ASR system by making selective use of existing speech databases.

An overview to previous works on system portability selected from literature was given in Section 1.5. Instead of building a new ASR application from scratch,

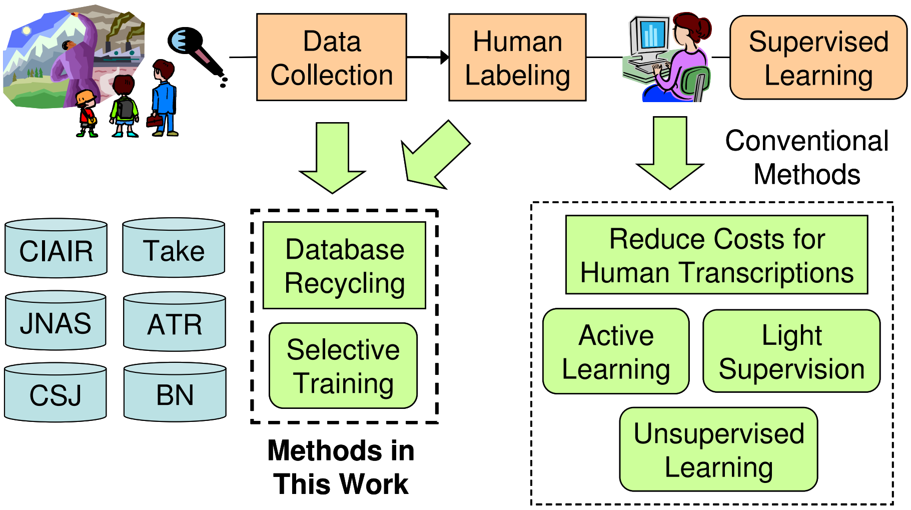

Figure 1.6. Classification of Current Work.

the reuse of existing data, models or whole systems is also worth considering. Therefore, topics such as investigation of cross-task performance, multi-source training and task adaptation are also considered.

The portability issue and the effect of data reuse in case of a real-environment ASR application is investigated in Chapter 3. The relationship between the development period and system performance is investigated first by conducting a development simulation for the speech-oriented guidance system *Takemaru* in Section 3.3. System performance is evaluated every month by several performance indicators for each system component. The amount of training data and the length of the development period until performance improvement stagnates are investigated.

Since from-scratch development of the *Takemaru* prototype system turns out to be very expensive, it is imperative to consider reuse of *Takemaru* for different users and a different environment. The goal is to achieve cost reduction when the task and/or domain of the system depending on the users and the environment changes. Therefore, the portability of the *Takemaru* system is investigated for the *Kita* environment in Section 3.4. Instead of constructing the *Kita* systems from scratch, reuse of *Takemaru* data and models and their update (adaptation) with *Kita* data is considered. From-scratch development is compared to reuse and update of the *Takemaru* system with respect to ASR and Q&A performance. Finally, the level of performance of the *Takemaru* and *Kita* systems is compared in case of system reuse, short-term, medium-term and long-term development.

In order to resolve the difficulty of conventional learning methods regarding task and domain-dependency, a general framework for data reuse and selective training is proposed in Chapter 4. Although the method is formulated for an HMM-based acoustic model, the method can also be applied to other problems and statistical models for which so-called sufficient statistics exist. In order to make this work as self-contained as possible, the technical background is explained in Chapter 2.

The proposed selective training algorithm is evaluated in Chapter 5 for human-transcribed and automatically transcribed speech data. It is investigated whether a preschool children and an elderly model can be constructed by selecting a subset from school children and adult speech data, respectively. Finally, the combination of unsupervised and selective training for semi-automatic construction of separate adult/child models for a real-environment ASR application is investigated.

Section 6 gives a summary and practical advice for cost-effective development of ASR applications and future directions.

Chapter 2

Fundamentals

This section gives an introduction to the fundamentals necessary in order to understand the technical details of this work. It is assumed that the reader is familiar with basic calculus, set theory and probability theory. Major references are [9, 62, 61, 51, 36, 18, 70].

2.1 Statistics

A set of elements $\{X_1, X_2, \ldots\}$ of the same properties, e.g. a series of measurements for a physical quantity, which should be investigated for certain characteristics is called the **population**. A subset of the population $\mathcal{X} = \{X_1, X_2, \ldots, X_n\}$ is called **sample**. The sample may also be infinite. Each element X_i is a random variable. Actual observations $\{x_1, x_2, \ldots, x_n\}$ are called **sample points** or **sample values**. If the random variables X_i are statistically independent, i.e. each element X_i is drawn from the underlying population with the same probability, and identically distributed, i.e the probability density functions $p(X_i)$ are identical for all i, $\mathcal{X}$ is called a **random sample**.

A **statistic** is a function of a given sample. Statistics are employed to estimate certain properties of the population. For example, the functions

$$g_1(\mathcal{X}) = \frac{1}{n}\sum_{i=1}^{n} X_i, \qquad g_2(\mathcal{X}) = \frac{1}{n}\sum_{i=1}^{n} \left[X_i - g_1(\mathcal{X})\right]^2 \tag{2.1}$$

of sample $\mathcal{X}$ are called **mean** and **variance**, respectively.

2.2 Learning Theory and Model Estimation

The purpose of learning is to estimate the underlying distribution $p(X)$ of the population X given a random sample $\mathcal{X}$. Since the number of possible distributions over a continuous sample space are uncountably infinite in general, assumptions are required for successful learning. A possible assumption would be that the sample values were generated by some **parametric density function** $f(x|\mathbf{\Theta})$, for example a **Gaussian density** with the parameters $\mathbf{\Theta} = (\mu, \sigma)$.

$$f(x|\mathbf{\Theta}) = \mathcal{N}(x|\mu, \sigma) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left[-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2\right] \tag{2.2}$$

The function $f(x|\mathbf{\Theta})$ is called learning machine or **model**. The goal of learning is obtain an estimate $\hat{\mathbf{\Theta}}$ for the value of the parameters $\mathbf{\Theta}$. It is important to select a function which is expected to match the true underlying distribution of the population. In the following methods for estimating model parameters are described.

2.2.1 Maximum Likelihood Estimation

Suppose a given random sample $\mathcal{X}$ was generated according to a certain probability density function of known type. However, the parameters itself are unknown and there is no prior knowledge about the range or probability of the parameters $\mathbf{\Theta}$. The idea of maximum likelihood estimation is to determine the parameter values which maximize the **log-likelihood function** $\log p(\mathcal{X}|\mathbf{\Theta})$ of the random sample.

$$\log p(\mathcal{X}|\mathbf{\Theta}) = \log \prod_{i=1}^{n} p(x_i|\mathbf{\Theta}) = \sum_{i=1}^{n} \log p(x_i|\mathbf{\Theta}) \tag{2.3}$$

The likelihood function $p(\mathcal{X}|\mathbf{\Theta})$ can be written as the product of the sample values likelihoods because observations are assumed to be statistically independent. If the density function $p(x|\mathbf{\Theta})$ is differentiable, a solution can be derived by using differential calculus. The derivation of the log-likelihood function by the parameter vector $\mathbf{\Theta} = (\Theta_1, \Theta_2, \ldots, \Theta_k)$ can be written using the nabla operator as

$$\boldsymbol{\nabla}_{\mathbf{\Theta}} \log p(\mathcal{X}|\mathbf{\Theta}) = \sum_{i=1}^{n} \boldsymbol{\nabla}_{\mathbf{\Theta}} \log p(x_i|\mathbf{\Theta}). \tag{2.4}$$

The necessary condition for a maximum is that all derivatives of the log-likelihood function are equal to zero.

$$\frac{\partial}{\partial\Theta_1} \sum_{i=1}^{n} \log p(x_i|\mathbf{\Theta}) = 0 \quad \cdots \quad \frac{\partial}{\partial\Theta_k} \sum_{i=1}^{n} \log p(x_i|\mathbf{\Theta}) = 0 \tag{2.5}$$

This system of equations has to be solved for the parameters $\mathbf{\Theta}$. If there are several solutions, a test for global maximum is required.

Example Let us determine the maximum-likelihood (ML) estimate in case of the Gaussian density (Eq. 2.2). The derivations of the log-likelihood function are

$$\frac{\partial}{\partial\mu} p(\mathcal{X}|\mathbf{\Theta}) = \frac{\partial}{\partial\mu} \sum_{i=1}^{n} \log \frac{1}{\sigma\sqrt{2\pi}} \exp\left[-\frac{1}{2}\left(\frac{x_i-\mu}{\sigma}\right)\right]^2 = \sum_{i=1}^{n} \left(\frac{x_i-\mu}{\sigma}\right), \tag{2.6}$$

$$\frac{\partial}{\partial \sigma} p(\mathcal{X}|\boldsymbol{\Theta}) = \frac{\partial}{\partial \sigma} \sum_{i=1}^{n} - \log \sigma \sqrt{2\pi} - \frac{1}{2} \left(\frac{x_i - \mu}{\sigma} \right)^2 = \sum_{i=1}^{n} \left(-\frac{1}{\sigma} + \frac{x_i - \mu}{\sigma^3} \right). \quad (2.7)$$

By setting all derivations to zero and solving the system for μ and σ^2, it is verified easily, that mean and variance from Eq. 2.1 are obtained as ML solution.

2.2.2 Sufficient Statistics

A statistic $\boldsymbol{s} = \phi(\mathcal{X})$ is called sufficient for parameter $\boldsymbol{\Theta}$ if the likelihood function $p(\mathcal{X}|\boldsymbol{\Theta})$ can be written as a function of $\boldsymbol{s}$ independent of parameter $\boldsymbol{\Theta}$.

$$p(\mathcal{X}|\boldsymbol{\Theta}) = p(\mathcal{X}|\boldsymbol{\Theta}, \boldsymbol{s}) = p(\mathcal{X}|\boldsymbol{s}) \quad (2.8)$$

The independence of $\boldsymbol{s}$ is satisfied, if the likelihood function can be factorized into a product of the form

$$p(\mathcal{X}|\boldsymbol{\Theta}) = g(\boldsymbol{s}, \boldsymbol{\Theta}) h(\mathcal{X}). \quad (2.9)$$

$g(\boldsymbol{s}, \boldsymbol{\Theta})$ is called **kernel density**. For example, the factorization of the likelihood function in case of a Gaussian density is given by

$$p(\mathcal{X}|\boldsymbol{\Theta}) = \left(\prod_{i=1}^{n} \frac{1}{\sqrt{2\pi\sigma^2}} \right) \exp \left[\frac{n}{2\sigma^2} \left(\mu^2 + s_2 - 2\mu s_1 \right) \right] \quad (2.10)$$

with the sufficient statistics $\boldsymbol{s} = (s_1, s_2)$

$$s_1 = \frac{1}{n} \sum_{i=1}^{n} x_i, \qquad s_2 = \frac{1}{n} \sum_{i=1}^{n} x_i^2 \quad (2.11)$$

The kernel density is given by

$$[g(\boldsymbol{s}, \boldsymbol{\Theta})]^{\frac{1}{n}} = \frac{1}{\sqrt{2\pi\sigma^2}} \exp \left[\frac{1}{2\sigma^2} \left(\mu^2 + s_2 - 2\mu s_1 \right) \right]. \quad (2.12)$$

A further, information-theoretic definition of sufficient statistics is that the mutual information $I(\boldsymbol{\Theta}; \mathcal{X})$ of the observed data and the parameter is identical to the mutual information $I(\boldsymbol{\Theta}; \boldsymbol{s})$ of the sufficient statistic and the parameter.

2.2.3 Expectation-Maximization Algorithm

In case of a more complex model, e.g. Gaussian Mixture Model (GMM) or Hidden Markov Model (HMM), the estimation of unknown parameters using the ML principle is analytically no longer possible. This is due to **hidden variables** (parameters), e.g. mixture index for GMM and state sequence for HMM, which cannot be observed directly. The Expectation-Maximization (EM) algorithm [16] is a framework for iteratively finding a ML solution in case of hidden variables.

However, the EM solution may be a local optimum of the likelihood function. Consequently, the initialization is very important.

If ML estimation as described in Section 2.2.1 is applicable to the considered model, the parameters $\boldsymbol{\Theta}^*$ corresponding to the global optimum of the likelihood function

$$\boldsymbol{\Theta}^* = \arg \max_{\boldsymbol{\Theta}} \; p(\mathcal{X}|\boldsymbol{\Theta}) \tag{2.13}$$

are determined. On the other hand, the idea of the EM algorithm is to start with some initial guess for the model parameters $\boldsymbol{\Theta}^{(0)}$ and then iteratively improve the estimate so that the likelihood increases, i.e.

$$p(\mathcal{X}|\boldsymbol{\Theta}^{[0]}) < p(\mathcal{X}|\boldsymbol{\Theta}^{[1]}) < p(\mathcal{X}|\boldsymbol{\Theta}^{[2]}) < \cdots < p(\mathcal{X}|\boldsymbol{\Theta}^{[j]}). \tag{2.14}$$

However, it is often difficult to use the likelihood function for optimization. Instead an auxiliary function Q can be employed which is defined as

$$Q(\boldsymbol{\Theta}^{[j+1]}|\boldsymbol{\Theta}^{[j]}) = \sum_{\boldsymbol{y}} p(\boldsymbol{y}|\mathcal{X}, \boldsymbol{\Theta}^{[j]}) \log p(\mathcal{X}, \boldsymbol{y}|\boldsymbol{\Theta}^{[j+1]}), \tag{2.15}$$

where $\boldsymbol{y}$ denotes the hidden variables. Using the information inequality

$$D(p(x|\boldsymbol{\Theta}) \parallel p(x|\boldsymbol{\Theta}')) = \sum_{x} p(x|\boldsymbol{\Theta}) \log \frac{p(x|\boldsymbol{\Theta})}{p(x|\boldsymbol{\Theta}')} \geq 0, \tag{2.16}$$

where $D(p(x|\boldsymbol{\Theta}) \parallel p(x|\boldsymbol{\Theta}'))$ is known as relative entropy, Kullback-Leibler (KL) distance or KL divergence, it can be shown that if the Q function increases, i.e. $Q(\boldsymbol{\Theta}^{[j+1]}|\boldsymbol{\Theta}^{[j]}) > Q(\boldsymbol{\Theta}^{[j]}|\boldsymbol{\Theta}^{[j]})$, the likelihood function also increases, i.e. $p(\mathcal{X}|\boldsymbol{\Theta}^{[j+1]}) > p(\mathcal{X}|\boldsymbol{\Theta}^{[j]})$ and vice versa. This is the key to the EM algorithm which works as follows:

I. Initialize model parameters $\boldsymbol{\Theta}^{(0)}$ and iteration index $j \leftarrow 0$

II. Repeat until $|Q(\boldsymbol{\Theta}^{[j]}|\boldsymbol{\Theta}^{[j]}) - Q(\boldsymbol{\Theta}^{[j+1]}|\boldsymbol{\Theta}^{[j]})| < \epsilon$ for some threshold $\epsilon > 0$

   1. E-Step: Calculate auxiliary function $Q(\boldsymbol{\Theta}|\boldsymbol{\Theta}^{[j]})$
   2. M-Step: Determine $\boldsymbol{\Theta}^{[j+1]} = \arg \max_{\boldsymbol{\Theta}} \; Q(\boldsymbol{\Theta}|\boldsymbol{\Theta}^{[j]})$
   3. Increase iteration index $j \leftarrow j + 1$

The maximization of Q can be performed by using differential calculus as demonstrated for the likelihood function. The EM algorithm will be applied to estimate the parameters of a hidden Markov model in the following section.

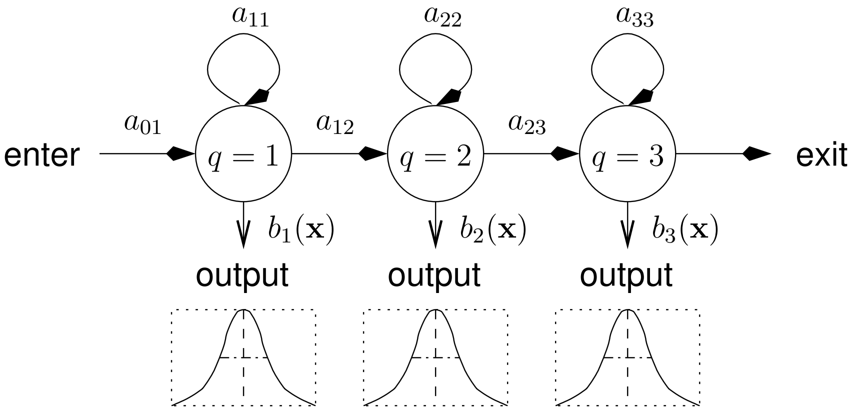

Figure 2.1. Illustration of left-to-right HMM with output densities.

2.3 Hidden Markov Model

A Hidden Markov Model (HMM) is a generative stochastic model for representing an observation sequences varying in time length. It models a two-stage stochastic process consisting of the hidden state sequence $\boldsymbol{s} = (s_1, s_2, \ldots, s_T)$ and the observation sequence $\boldsymbol{X} = (\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_T)$. State transitions and observations only depend on the current state. Figure 2.1 depicts a left-to-right HMM typical for modeling subword units in speech recognition. A HMM can be completely described by a **transition probability** matrix $\boldsymbol{A} = (a_{q'q})$ and parameters of the **output probability** density $b_q(\boldsymbol{x})$. The **Gaussian Mixture Model** (GMM) is often employed as output density because it can approximate arbitrary sample distributions and is mathematically easy tractable. A multivariate Gaussian is given by

$$\mathcal{N}(\boldsymbol{x}|\boldsymbol{\mu}, \boldsymbol{\Sigma}) = \frac{1}{\sqrt{(2\pi)^d|\boldsymbol{\Sigma}|}} \exp\left[-\frac{1}{2}(\boldsymbol{x} - \boldsymbol{\mu})^T\boldsymbol{\Sigma}^{-1}(\boldsymbol{x} - \boldsymbol{\mu})\right], \tag{2.17}$$

where d denotes the dimension of the feature space. A Gaussian mixture density can then be defined as

$$b_q(\boldsymbol{x}) = \sum_{m=1}^{M} b_{qm}(\boldsymbol{x}) = \sum_{m=1}^{M} w_{qm}\, \mathcal{N}(\boldsymbol{x}|\boldsymbol{\mu}_{qm}, \boldsymbol{\Sigma}_{qm}) \tag{2.18}$$

with the **mixture weights** w_m. M is the number of mixture components. $\boldsymbol{\Theta}$ denotes the complete set of HMM parameters in the following.

There are three basic issues involved with HMMs. The problem of (1) probability calculation, (2) parameter estimation problem and (3) decoding. In detail:

1. Calculate the output probability $p(\boldsymbol{X}|\boldsymbol{\Theta})$ given observation $\boldsymbol{X}$
2. Estimate ML parameters $\hat{\boldsymbol{\Theta}}^* = \arg\max_{\boldsymbol{\Theta}}\ p(\boldsymbol{X}|\boldsymbol{\Theta})$ given training data $\boldsymbol{X}$
3. Determine the best state sequence $\boldsymbol{s}^* = \arg\max_{\boldsymbol{s}}\ p(\boldsymbol{s}|\boldsymbol{X}, \boldsymbol{\Theta})$ given $\boldsymbol{X}$

Probability Calculation To calculate the probability of an observation, all possible state sequences of the HMM have to be considered.

$$p(\boldsymbol{X}|\boldsymbol{\Theta}) = \sum_{\boldsymbol{s}} p(\boldsymbol{X}, \boldsymbol{s}|\boldsymbol{\Theta}) = \sum_{\boldsymbol{s}} \left[\prod_{t=1}^{T} a_{s_{t+1}s_t}\right] \left[\prod_{t=1}^{T} b_{s_t}(\boldsymbol{x}_t)\right] \tag{2.19}$$

However, directly calculating this equation by complete enumeration of all possible state sequences is computationally too expensive. An effective method to calculate the observation probability is the employment of the forward or backward probabilities. Their definition is:

- Forward probability: $\alpha_q(t) = p(\boldsymbol{x}_1 \cdots \boldsymbol{x}_t, s_1 \cdots s_t, s_t = q|\boldsymbol{\Theta})$
- Backward probability: $\beta_q(t) = p(\boldsymbol{x}_{t+1} \cdots \boldsymbol{x}_T, s_{t+1} \cdots s_T, s_t = q|\boldsymbol{\Theta})$

Let $\mathcal{Q}$ be the set of all HMM states. $\mathcal{Q}_0$ and $\mathcal{Q}_e$ denote the set of initial and final states, respectively. The **forward algorithm** to calculate $p(\boldsymbol{X}|\boldsymbol{\Theta})$ is

1. Initialize: $\alpha_{q\in\mathcal{Q}_0}(0) \leftarrow 1, \quad \alpha_{q\in\mathcal{Q}\setminus\mathcal{Q}_0}(0) \leftarrow 0, \quad t \leftarrow 0$
2. Repeat until last observation reached:

   a. Go to next observation: $t \leftarrow t+1$

   b. Update probabilities: $\alpha_r(t) = \left[\sum_{q\in\mathcal{Q}} \alpha_q(t-1)a_{qr}\right] b_r(\boldsymbol{x}_t), \quad r \in \mathcal{Q}$
3. Return observation probability: $p(\boldsymbol{X}|\boldsymbol{\Theta}) = \sum_{q\in\mathcal{Q}_e} \alpha_q(T)$

The formulation of the time-reversed **backward algorithm** is analogue to the forward algorithm using backward probabilities.

Parameter Estimation That the direct maximization of the likelihood $p(\boldsymbol{X}|\boldsymbol{\Theta})$ is not possible due to hidden variables has already been discussed. Therefore, the optimization of the auxiliary Q function

$$Q(\hat{\boldsymbol{\Theta}}|\boldsymbol{\Theta}) = \sum_{\boldsymbol{z}} p(\boldsymbol{z}|\boldsymbol{X}, \boldsymbol{\Theta}) \log p(\boldsymbol{z}, \boldsymbol{X}|\hat{\boldsymbol{\Theta}}) \tag{2.20}$$

is considered, where $\boldsymbol{z} = ((s_1, l_1), \ldots, (s_T, l_T))$ denotes the state and mixture index sequence. When substituting HMM transition and output probabilities we obtain

$$= \sum_{\boldsymbol{z}} \frac{P(\boldsymbol{z}, \boldsymbol{X}|\boldsymbol{\Theta})}{P(\boldsymbol{X}|\boldsymbol{\Theta})} \log \left[\prod_{t=1}^{T} \hat{a}_{s_{t-1}s_t}\right] \left[\prod_{t=1}^{T} \hat{b}_{s_t l_t}(x_t)\right]. \tag{2.21}$$

By defining the state transition probability as

$$\gamma_{q'\to q}(t) = \frac{1}{P(\boldsymbol{X}|\boldsymbol{\Theta})} \sum_{\boldsymbol{z}} P(\boldsymbol{X}, s_{t-1} = q', s_t = q|\boldsymbol{\Theta}) \tag{2.22}$$

and state and mixture occupation probability as

$$\gamma_{qm}(t) = \frac{1}{P(\boldsymbol{X}|\boldsymbol{\Theta})} \sum_{\boldsymbol{z}} P(\boldsymbol{X}, s_t = q|\boldsymbol{\Theta}) \left[\frac{b_{qm}(\boldsymbol{x}_t)}{\sum\limits_n b_{qn}(\boldsymbol{x}_t)} \right], \tag{2.23}$$

the Q function can be transformed to be proportional to

$$\propto \sum_{q'} \sum_{q} \sum_{t} \gamma_{q' \to q}(t) \log \hat{a}_{q'q} + \sum_{q} \sum_{m} \sum_{t} \gamma_{qm}(t) \log \hat{w}_{qm} \; \mathcal{N}(\boldsymbol{x}_t|\hat{\boldsymbol{\mu}}_{qm}, \hat{\Sigma}_{qm}). \tag{2.24}$$

If given fixed, initial parameters $\boldsymbol{\Theta}$ and the constraints

$$\sum_q \hat{a}_{q'q} = 1, \qquad \sum_m \hat{w}_{qm} = 1, \qquad \int_{\boldsymbol{x}} \mathcal{N}(\boldsymbol{x}|\hat{\boldsymbol{\mu}}_{qm}, \hat{\boldsymbol{\Sigma}}_{qm}) d\boldsymbol{x} = 1 \tag{2.25}$$

on transition probabilities, mixture weights and output densities, it can be shown that the estimates for transition probabilities $\hat{a}_{q'q}$, mixture weights $\hat{w}_{qm}$, Gaussian mean vectors $\hat{\boldsymbol{\mu}}_{qm}$ and Gaussian covariance matrices $\hat{\boldsymbol{\Sigma}}_{qm}$ maximizing the Q function are

$$\hat{a}_{q'q} = \frac{\sum\limits_t \gamma_{q' \to q}(t)}{\sum\limits_q \sum\limits_t \gamma_{q' \to q}(t)}, \tag{2.26}$$

$$\hat{w}_{qm} = \frac{\sum\limits_t \gamma_{qm}(t)}{\sum\limits_n \sum\limits_t \gamma_{qn}(t)}, \tag{2.27}$$

$$\hat{\boldsymbol{\mu}}_{qm} = \frac{\sum\limits_t \gamma_{qm}(t) \; \boldsymbol{x}_t}{\sum\limits_t \gamma_{qm}(t)}, \tag{2.28}$$

$$\hat{\boldsymbol{\Sigma}}_{qm} = \frac{\sum\limits_t \gamma_{qm}(t) \; (\boldsymbol{x}_t - \hat{\boldsymbol{\mu}}_{qm})(\boldsymbol{x}_t - \hat{\boldsymbol{\mu}}_{qm})^T}{\sum\limits_t \gamma_{qm}(t)}. \tag{2.29}$$

The state transition $\gamma_{q' \to q}(t)$ and state occupation probabilities $\gamma_{qm}(t)$ can be calculated effectively using forward $\alpha_q(t)$ and backward $\beta_q(t)$ probabilities.

$$\gamma_{q' \to q}(t) = \frac{\alpha_{q'}(t-1) a_{q'q} b_q(\boldsymbol{x}_t) \beta_q(t)}{\sum\limits_r \alpha_r(t) \beta_r(t)} \tag{2.30}$$

$$\gamma_{qm}(t) = \frac{\sum\limits_r \alpha_r(t-1) a_{rq} b_{qm}(\boldsymbol{x}_t) \beta_q(t)}{\sum\limits_r \alpha_r(t) \beta_r(t)} \tag{2.31}$$

Best State Sequence The best state sequence $\boldsymbol{s}^*$ with maximum probability $p(\boldsymbol{s}^*|\boldsymbol{X}, \boldsymbol{\Theta})$ can be determined easily by modifying the forward algorithm. Instead of summing over all forward probabilities of the last time frame, only the state transition with probability so far is considered. Furthermore it is important to additionally remember the index of the previous state of the partial path with maximum probability. For notational simplicity we assume there are only one initial q_0 and one final state q_e.

1. Initialize: $t \leftarrow 0, \quad c_{q_0}(0) \leftarrow 1, \quad d_{q_0}(0) = q_0$
2. Repeat until last observation reached:
   a. Go to next observation: $t \leftarrow t+1$
   b. Update probabilities: $c_r(t) = \max\limits_{q \in \mathcal{Q}} c_q(t-1) a_{qr} b_r(\boldsymbol{x}_t), \quad r \in \mathcal{Q}$
   c. Update state indices: $d_r(t) = \arg\max\limits_{q \in \mathcal{Q}} \; c_q(t-1) a_{qr} b_r(\boldsymbol{x}_t), \quad r \in \mathcal{Q}$
3. Return probability and indices of best state sequence $\boldsymbol{s}^*$
   a. $p(\boldsymbol{s}^*|\boldsymbol{X}, \boldsymbol{\Theta}) = c_{q_e}(T)$
   b. $\boldsymbol{s}^* = (q_0, \ldots, d_{d_{d_{q_e}(T)}(T-1)}(T-2), d_{d_{q_e}(T)}(T-1), d_{q_e}(T), q_e)$

2.4 Automatic Speech Recognition

In the following a brief introduction to automatic speech recognition is given. Representation of speech at different levels, feature extraction, HMM-based acoustic model, types of acoustic models, N-gram language model, their relationship and the decoder are explained.

2.4.1 Speech and Transcription

There are several possibilities to express a human's utterance in written form. The way we are most used to is to write down the spoken word sequence. However, there are other and more well defined levels and units which are described in the following.

Phoneme Humans can produce speech sounds by sending air pressure waves from the lung through the trachea, pulsating the glottis and moving their speech organs. Since the lips, tongue and other articulators are moving continuously it is actually difficult to separate speech sounds from each other in order to identify certain speech units. Single speech sounds are called phones. A class of phones which are the smallest units to differentiate the meaning of words is called phoneme. Phonemes are language specific. The phonemes of the Japanese language are given in Table 2.1.

Table 2.1. Phoneme set of the Japanese language

Vowels	Consonants
a i u e o	k g s z j t d m n h f b p y r w q
a: i: u: e: o:	sh ts ch dy gy by py ky ny my hy N

Table 2.2. Mapping of selected Katakana to phonemes

Katakana Character → Phoneme Sequence									
ア	a	イ	i	ウ	u	エ	e	オ	o
カ	k a	キ	k i	ク	k u	ケ	k e	コ	k o
サ	s a	シ	sh i	ス	s u	セ	s e	ソ	s o
タ	t a	チ	ch i	ツ	ts u	テ	t e	ト	t o
ナ	n a	ニ	n i	ヌ	n u	ネ	n e	ノ	n o
ハ	h a	ヒ	h i	フ	f u	ヘ	h e	ホ	h o
マ	m a	ミ	m i	ム	m u	メ	m e	モ	m o
ヤ	y a			ユ	y u			ヨ	y o
ラ	r a	リ	r i	ル	r u	レ	r e	ロ	r o
ワ	w a							ヲ	o
ン	N								

Grapheme Spoken units can be expressed in visual form as graphemes. There are different kinds of graphemes depending on the language. Many languages like English, German, French, Spanish use letters as graphemes. The letters are speech sound oriented although automatic conversion of graphemes into phonemes and vice versa is not trivial in most cases. Chinese and Japanese employ so-called Kanji characters which are meaning-oriented graphemes. Apart from Kanji there are two other kinds of grapheme systems in Japanese: Katakana and Hiragana called Kana together (Table 2.3). Although meaning ambiguities may arise in practice, it is possible to write any Japanese sentence using only Kana. Kana have the nice property that there is a direct mapping from Kana graphemes to phonemes. The mapping is shown in Table 2.2 for selected graphemes.

Morpheme The smallest unit within a language which carries a meaning is called morpheme. A morpheme can either be expressed phonemically as well as graphemically. Japanese is a language with no clear definitions of words as in English or in German because there are no word boundaries. In order to cut

Table 2.3. Examples of Japanese graphemes

Hiragana	あ い う え お か き く け こ さ し す せ そ た ち つ て と
Katakana	ア イ ウ エ オ カ キ ク ケ コ サ シ ス セ ソ タ チ ツ テ ト
Kanji	日 本 人 上 下 石 土 月 水 火 木 金 羽 鳥 家 住 休 池 魚 肉

Table 2.4. Examples for Japanese morpheme sequences

どこ — で — 生まれ — まし — た
将来 — に — 何 — に — なり — たい — です — か
近く — に — レストラン — が — あり — ます — か

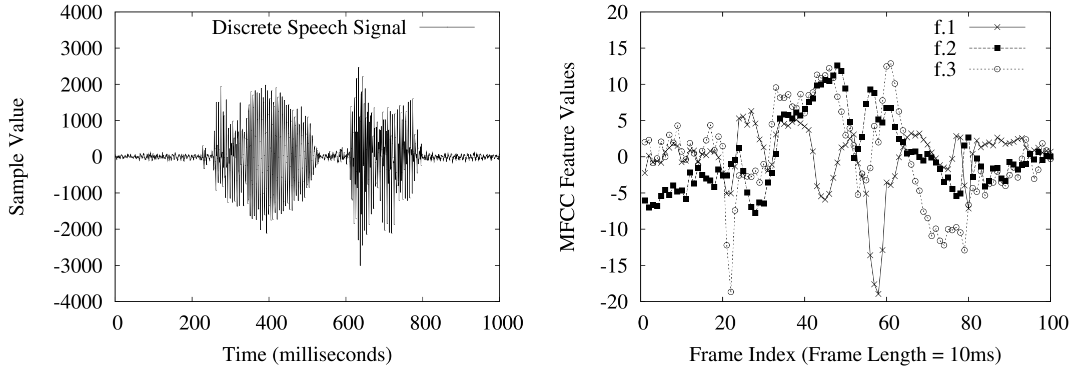

Figure 2.2. Left: Speech signal for a Japanese utterance with the phone sequence /k o N n i ch i w a/. Right: The first three components of the corresponding MFCC feature vector sequence.

a sentence into smaller peaces it can be separated into morphemes. A freely available morphological analyzer for Japanese is Chasen [10]. Each morpheme determined by Chasen consists of three parts: semantic representation using Kana and Kanji, the reading using Katakana and a number code indicating parts of speech, conjugation type and conjugation form. Examples for the semantic part of morpheme sequences are shown in Table 2.4. In this work the terms word and morpheme will be used as synonyms.

2.4.2 Feature Extraction

The air pressure waves produced by a human who is currently speaking can be sensed by a microphone and converted into an analogue signal. The analogue signal is an electric current of changing voltage depending on the air wave's intensity. The analogue signal can be digitized by sampling every fixed time interval and quantizing the amplitude level on a discrete scale. The outcome is a discrete speech signal.

However, a time-domain representation of speech is not suitable for recognition. Therefore, methods for extracting information relevant to speech recognition have been developed. MFCC and PLP are widely accepted as features for speech recognition. Figure 2.2 shows a discrete speech signal and the first components of the MFCC feature vector.

The processing steps to obtain a short-term spectrum from the discrete speech signal are shown in Figure 2.3. First a pre-emphasis filter is applied to enhance

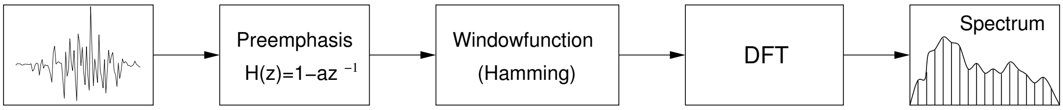

Figure 2.3. Conversion of speech signal from time-domain to frequency domain.

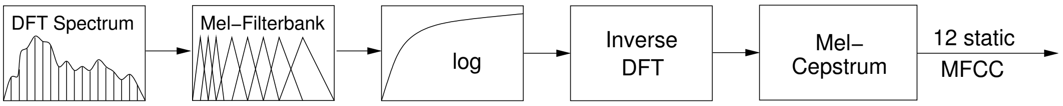

Figure 2.4. Conversion of short-term spectral coefficients to mel-frequency cepstrum coefficients.

higher frequency components. Each fixed time-interval (10-20 ms are common) a speech frame of 20-30 ms is considered for feature extraction. To obtain a discrete power spectrum the discrete Fourier transformation (DFT) is employed. Since the premise for DFT is a periodic signal, a window function, e.g. hamming, is applied to assure a smooth transition at boundaries.

The short-term Fourier spectrum (absolute values) is transformed with a mel-filterbank. Triangular filters with a spacing based on the mel-frequency scale are employed for critical band integration. The critical bands have a psychoacoustic equivalent in human perception of speech sounds. After taking the logarithm or root of mel-frequency coefficients in order to compensate for perceptual intensity (loudness), the inverse Fourier transform is applied to obtain mel-frequency cepstrum coefficients (MFCCs). By carefully selecting the range of mel-frequency coefficients, it is possible to separate the harmonic structure of the speech signal, which is due to the excitation pulses from the glottis, from the vocal tract characteristics. Only the latter information is important for speech recognition. In practice up to 12 MFCCs are employed.

In addition to the static cepstrum coefficients just described, log-energy E and the first derivation of static MFCCs and the log-energy are often employed. These dynamic features (also called delta coefficients) can be calculated efficiently by using the slope of the regression line over five speech frames, i.e. the current frame and two preceding and following frames each. The acoustic feature vector employed for all experiments in this thesis consists of 12 MFCCs, 12 Δ MFCCs and Δ Energy. A good survey about feature extraction for ASR is [59].

2.4.3 HMM-based Acoustic Model

The acoustic model consist of models for words or **subword units** of the considered target language. Example for subword units are syllables and phonemes. Subword units should be general, i.e. any possible word can be synthesized. The larger the subword unit, the more detailed is the modeling of acoustic characteristics. However, it is difficult to obtain enough training data if the subword units are large. Unless the application is digit or command recognition, phoneme-

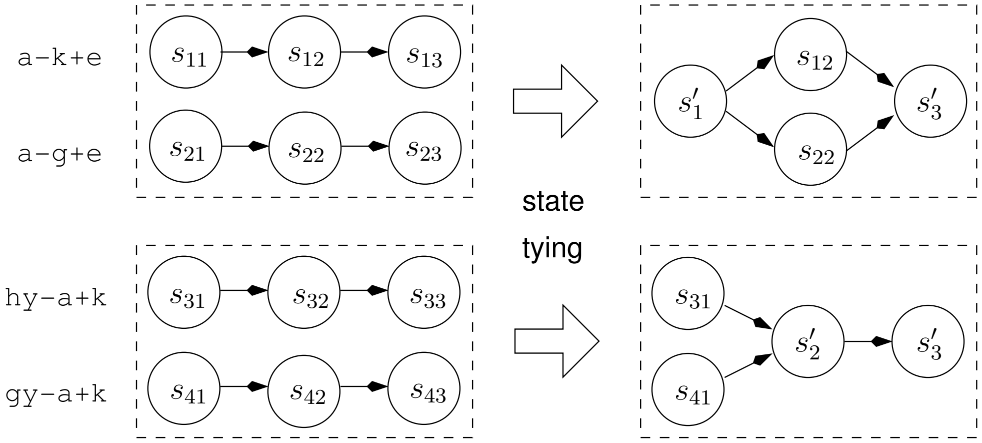

Figure 2.5. Example for acoustic model with state-tied triphone HMMs.

based subword units are used most widely. In the following three kinds of acoustic models are considered. They are all based on phonemes.

Context-Independent Monophone Each phoneme is modeled exclusively by one HMM as shown in Figure 2.1. Since there are 40 phonemes in Japanese (Table 2.1), 40 HMMs are needed. Furthermore, it is necessary to model the beginning and ending silence of utterances and short pauses between words and/or sentences. A monophone acoustic model consists of 43 HMMs. The output density of each HMM state is modeled by a Gaussian mixture model. In order to keep the number of model parameters low, it is common to use only diagonal covariance matrices. If the components of the input feature vector are decorrelated, e.g. by applying a PCA transformation on the feature space, covariances can be neglected without losing information. The advantage of monophone models is that they can be constructed with only speech data and that they are quite robust against variabilities due to speaker characteristics and pronunciation. However, the disadvantage of context-independent models is that their performance is limited since phoneme contexts are not taken into account.

Context-Dependent Triphone In order to model coarticulation effects, i.e. the pronunciation of a phoneme is influenced by preceding and following phonemes, context-dependent subword units have been proposed. The idea of the triphone model is to employ a different model for each phoneme depending on the context. For example, the /k/ in /akai/ and /ikeda/ is enclosed by different vowels. Consequently, the /k/ of each word is modeled by two different HMMs `a-k+a` and `i-k+e`. The difficulty with triphones is that their number is very large: $40^3 = 64,000$. Even when eliminating practically irrelevant triphones more than 20,000 HMMs would have to be constructed. In order to reduce the number of parameters, similar HMM states are clustered to construct state-tied triphone models. An example for state clustering is shown in Figure 2.5. To

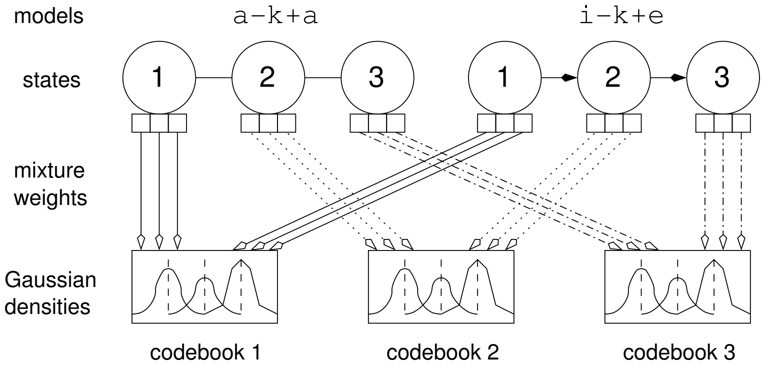

Figure 2.6. Structure of acoustic model with phonetically tied mixtures.

obtain a reasonable clustering result, constraints are employed such as clustering only states belonging to the same phoneme class, e.g. plosives, fricatives or nasals, and having the same state index. The result of clustering is that the same output density is assigned to a tied state which originally belonged to two different HMMs.

Phonetically Tied Mixture (PTM) An acoustic model with even more constraints to reduce the number of parameters and which has been developed for fast decoding with the speech recognition engine Julius [37] is the phonetically tied mixture model [45]. It is synthesized from the state information of a state-tied triphone model and the Gaussian mixture density codebooks of the monophone model. The codebook is shared among state of triphone models which have the center phoneme, e.g. triphones `a-k+a` and `i-k+e` have `k` in common, so they use the same density codebook. A further constraint is that the state index must also be the same. With all these constraints, it seems that PTM and monophone model are identical. The difference actually is that the mixture weights are not shared among different states. An illustration of PTM model structure is given in Figure 2.6.

2.4.4 Statistical Language Model

The purpose of the language model is to determine the sequences of words which a speech recognizer should be able to recognize. The most widely used statistical framework for language modeling is the N-**gram**. An N-gram is a sequence of n words.

In order to calculate the exact probability of a sentence, i.e. a word sequence $\boldsymbol{W} = (w_1, w_2, \ldots, w_n)$, the complete word history would have to be considered. Since this is infeasible in practice, the probability $P(\boldsymbol{W})$ is approximated using N-gram probabilities.

$$p(\boldsymbol{W}) = \prod_{i=1} p(w_i|w_1, \ldots, w_{i-1}) \approx \prod_{i=1} p(w_i|w_{i-N+1}, \ldots, w_{i-1}) \tag{2.32}$$

The N-gram probabilities are usually obtained from a large text corpus of the target domain by counting the frequency of N-grams. The ML estimate of N-gram probabilities is

$$p(w_i|w_{i-N+1}, \ldots, w_{i-1}) = \frac{\#(w_{i-N+1}w_{i-N+2} \cdots w_i)}{\#(w_{i-N+1}w_{i-N+2} \cdots w_{i-1})}. \tag{2.33}$$

There are practical difficulties with probability estimation. Even if the vocabulary size v is small, the theoretic number of possible $N = 3$-grams for $v = 1,000$ different words is $v^N = 1,000,000,000$. Although only practically relevant N-grams have to be considered, it is obvious that it is infeasible to provide enough domain-specific training data to estimate N-gram probabilities reliably. A further difficulty are so-called unseen N-grams which appear in the test but not in the training data.

Therefore, methods for smoothing N-gram probabilities and the back-off to lower-order $N-1$-grams, etc. have been developed. The purpose of smoothing is to assign a very small but reasonable amount of probability mass to N-grams with zero probability. This enables the probability calculation for unseen N-grams. Results of an empirical study of comparing many different smoothing techniques [12] showed that Kneser-Ney smoothing, an absolute discounting method yields the best estimates for N-gram probabilities and the best performance for speech recognition.

2.4.5 Decoder

The decoder is the core part of the speech recognizer. It uses information provided by the acoustic and language model to search for the optimal word sequence $\boldsymbol{W}$ given the feature vector sequence $\boldsymbol{X}$ of the input speech signal. The word sequence which maximizes the posterior probability $p(\boldsymbol{W}|\boldsymbol{X})$ is considered as optimal. Using the Bayes rule

$$p(\boldsymbol{W}|\boldsymbol{X}) = \frac{p(\boldsymbol{X}|\boldsymbol{W})p(\boldsymbol{W})}{p(\boldsymbol{X})}, \tag{2.34}$$

the decision rule based on the posterior probability can be written as

$$\boldsymbol{W}^* = \arg\max_{\boldsymbol{W}} \; p(\boldsymbol{X}|\boldsymbol{W})p(\boldsymbol{W}). \tag{2.35}$$

The acoustic likelihood $p(\boldsymbol{X}|\boldsymbol{W})$ can be calculated using the acoustic model. The prior probability of a word sequence $p(\boldsymbol{W})$ is provided by the language model. The evidence $p(\boldsymbol{X})$ can be omitted, since it is independent of the word sequence $\boldsymbol{W}$.

2.5 Evaluation Measures

2.5.1 OOV Rate and Perplexity

Out-Of-Vocabulary (OOV) rate and perplexity for a test data set disjoint from the training data set are two measures to evaluate the quality of the language model. The OOV rate is defined as

$$\text{OOV rate} = \frac{\#\text{ unknown words in test set}}{\#\text{ words in test set}}. \tag{2.36}$$

If the OOV rate of the test data is $X\%$, the maximum possible recognition rate is $100 - X\%$. The perplexity (PP) is defined as

$$\text{PP} = 2^L, \tag{2.37}$$

where the log-probability L is calculated as

$$L = -\frac{1}{n}\sum_{i=1}^{n}\log_2 p(w_i|w_1,\ldots,w_{i-1}) \approx -\frac{1}{n}\sum_{i=1}^{n}\log_2 p(w_i|w_{i-N+1},\ldots,w_{i-1}). \tag{2.38}$$

A perplexity $\text{PP} = K$ means that every word node in the recognition is followed by K word nodes on average. The higher the perplexity, the more difficult is the recognition task. Consequently, a language model is the better the lower the OOV rate and perplexity are.

2.5.2 Recognition Performance

There are several possibilities to evaluate the performance of a speech recognizer. The measures differ w.r.t. the errors which should be taken into account. There are insertion (#ins), substitution (#sub) and deletion (#del) errors. The correct rate ignores insertion errors and is obtained easily by counting the number of correct (#cor) tokens in the recognition hypothesis.

$$\text{Correct Rate} = \frac{\#\text{cor}}{\#\text{ tokens in reference}} \tag{2.39}$$

On the other, the accuracy takes all error kinds into account and is defined by

$$\text{Accuracy} = 1 - \frac{\#\text{sub} + \#\text{ins} + \#\text{del}}{\#\text{ tokens in reference}}. \tag{2.40}$$

Both measures can be defined for the phoneme as well as the word level.

Table 2.5. Assumed system development base costs and cost factors

Base costs	100,000	points
Collection of one utterance	1	point
Transcription of one utterance	24	points

2.5.3 Response Accuracy

The ASR application considered in this thesis is a speech-oriented guidance system. Rather than measuring speech recognition performance, the rate of user satisfaction is of more practical importance. However, user satisfaction is difficult to measure in practice, since it requires to interview each user whether he or she is satisfied by the system's response. Nevertheless, there is the possibility to measure an approximation of user satisfaction by providing a reference response for each user input. A second human observer subjectively assigns a possibly correct system response to a set of transcribed user inputs. The relative share of automatically derived system responses matching the human labeled response is defined as response accuracy, i.e.

$$\text{Response Accuracy} = \frac{\#\ \text{correctly answered utterances}}{\#\ \text{test utterances}}. \tag{2.41}$$

2.5.4 Development Costs

There are several aspects about the development costs of an ASR system. To collect task-specific speech data, a prototype system has to be operated in the target environment. This will usually cause costs for computer hardware, software, and system installation. Collection of a certain amount of data itself takes time. Therefore, not only the number of collected data but also the collection period has to be considered. Furthermore, costs for transcribing and labeling collected data by humans may arise. From this observation we obtain two basic indicators for development costs, (1) the number of collected task-specific data and (2) the number of human-transcribed task-specific data.

By weighing each indicator with a cost factor and adding base development costs for system installation, a system's total development costs can be calculated. However, it is difficult for a researcher of a public research institution to assess the real cost factors precisely. Considering costs for computer hardware, working hours and salary of technical assistants, the base costs and cost factors as shown in Table 2.5 are assumed.

2.6 Text Matching

The speech-oriented guidance system considered as ASR application is this work employs a database of question and answer pairs (Q&A DB) for response generation. The speech recognition result R is compared to the example question E_j of each Q&A pair. The response of the best matching pair is presented to the user.

The comparison of the recognition result $(r_1, \ldots, r_k)$ with an example question $(e_1, \ldots, e_l)$ can be carried out at the morpheme level. The morpheme sequence of each example question is treated as a template. The distance between the recognized morpheme sequence and the template can be calculated as the minimum edit-distance[18] using dynamic programming. The minimum edit-distance counts the number of insertions, deletions and substitutions based on the Viterbi alignment between the two morpheme sequences. A drawback of DP matching are its computational cost. $k * l$ morpheme comparisons would be required per example question.

To reduce computational costs a distance measure using the bag-of-words representation is employed. By ignoring the morpheme order of the recognition result and example question, the morpheme sets $R = \{r_1, \ldots, r_k\}$ and $E = \{e_1, \ldots, e_l\}$ are obtained. The distance between these two sets is defined by the number of morphemes in the intersection set divided by the number of elements in the larger set. The normalization assures that the distance value is in the interval $[0; 1]$. Furthermore, the n-best recognition hypotheses $R_1, \ldots, R_n$ can be taken into account by calculating the average of the n distances.

$$E^* = \arg\max_{E} \ \frac{1}{n} \sum_{i=1}^{n} \frac{|R_i \cap E|}{\max\{|R_i|, |E|\}} \tag{2.42}$$

This score can be calculated efficiently by providing a mapping from morphemes e to lists of example question indices j in which each morpheme is occurring.

Chapter 3

Data, Model and System Reuse

The idea to reuse existing speech data, models or a complete system to save developments costs of an ASR application seems to be straightforward and possible at all appearances. However, there is the problem of task and domain dependency of ASR systems as outlined in Section 1.3.1. Therefore, it has to be investigated in how far reuse of existing resources can shorten the development cycle and reduce development costs.

With a development simulation of an open-domain dialogue system for two different real environments, the cost performance is analyzed empirically by considering the relationship between development period, amount of human-transcribed training data and several performance indicators. After constructing a dialogue system prototype for one environment, the system's adaptation to a second environment is carried out. From a performance comparison of from-scratch development and reuse/update of the prototype system for the second environment, possibilities for development cost reduction become apparent (Figure 3.1).

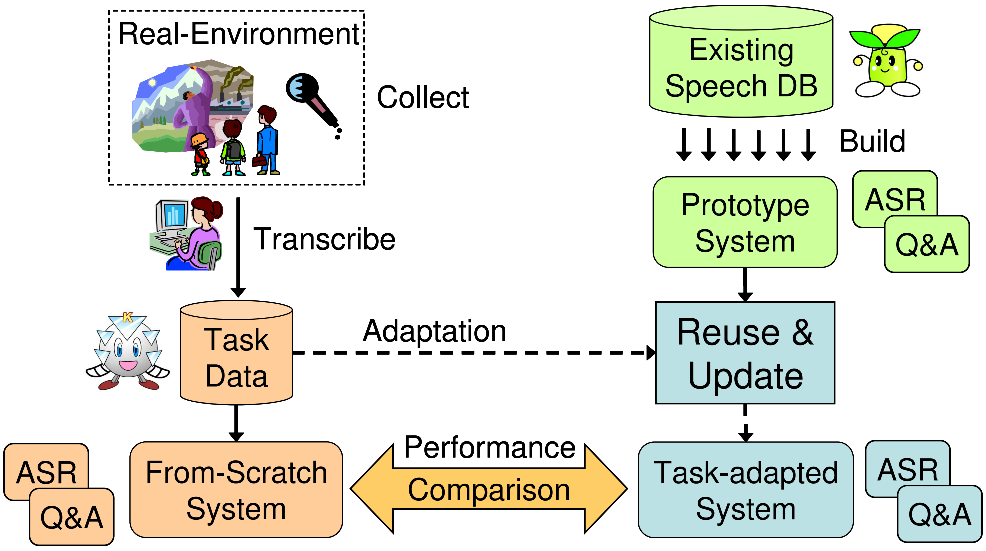

Figure 3.1. Task and Domain-Dependency of Real-Environment Applications.

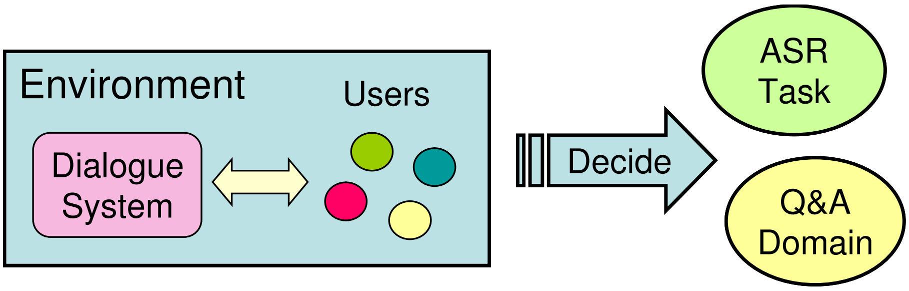

Figure 3.2. Task and Domain-Dependency of Real-Environment Applications.

This chapter is organized as follows: Section 3.1 gives a brief overview to spoken dialogue systems. Section 3.2 first gives an example of an open-domain dialogue system, a speech-oriented guidance system. Furthermore, three implementations of a guidance system, *Takemaru*, *Kita-chan* and *Kita-robo*, which have been installed and operated in a real environment for several years, and the corresponding real-environment speech databases are described. Results for a development simulation of *Takemaru* are given in Section 3.3. The effect of reusing the *Takemaru* system for the *Kita* environment is evaluated in Section 3.4. Finally, the results of a domain analysis and domain comparison of the *Takemaru* and the *Kita* systems are reported in Section 3.5.

3.1 Spoken Dialogue Systems

Spoken dialogue systems may be categorized into rather system-driven, goal-oriented systems and rather user-driven, open-domain, access-oriented systems. Examples for goal-oriented systems are flight reservation [68], train reservation [44] or bus information [63]. Their drawback is that the system's scope is most often defined by the developer ignoring completely the actual behavior of potential users. This is problematic, since users often do not behave as developers have expected it. Ignoring this circumstance will most often result in a poor system performance.

Access-oriented dialogue systems, e.g. for call routing [26], speech-activated text retrieval [34] or speech-oriented guidance [54] suffer less from this problem, since they are mainly user-driven. The user can formulate his request freely in natural language and can immediately obtain a response from the system after the first query. The system's domain is open by definition from the beginning, since the system's scope is determined by potential users and the system's environment (Figure 3.2). Therefore, system development only based on an engineer's ideas is unlikely to succeed. Consequently, it is imperative to collect real speech data in the target environment in order to build a system which can cope with a wide variety of actual user queries under realistic conditions.

An overview to development and portability issues of dialogue systems with a focus on language modeling and understanding can be found in [21]. Active and

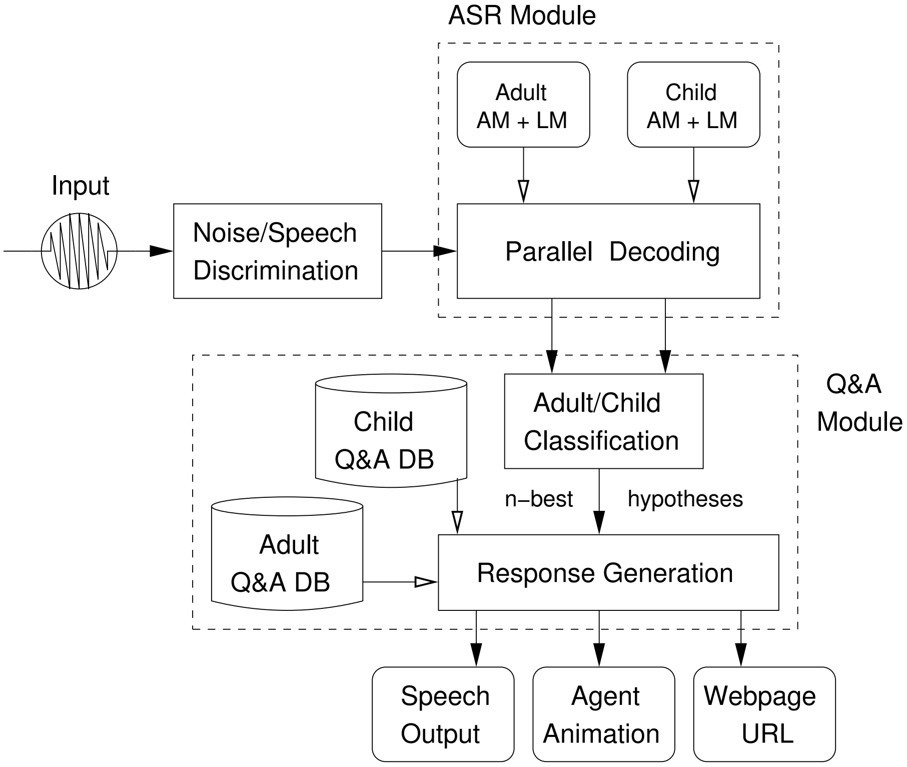

Figure 3.3. Main building blocks of a speech-oriented guidance system.

unsupervised learning are often proposed as cost-effective methods for developing and adapting the acoustic model of the speech recognizer. While active learning can reduce the costs for human-labeling of speech data without compromising the performance [38], unsupervised learning can already be outperformed even with limited amounts of human-labeled data [79]. Furthermore, the employment of text data from the web or external sources is a common approach to bootstrap language models if domain-specific data is not available. However, a domain-specific model which is only trained on few thousand sentences is likely to yield a higher performance [66].

These findings from previous research indicate that human-labeled, task-specific data are required for system development if performance cannot be compromised. Furthermore, reports on development, long-term operation and portability of an open-domain dialogue system considering all system components simultaneously are rare. Therefore, the purpose of this chapter is to investigate the amount of real data which should be collected to develop the ASR and Q&A components of a speech-oriented guidance system with reasonable performance. Secondly, it is investigated in how far components of a working prototype can be reused to build a second system for a different environment.

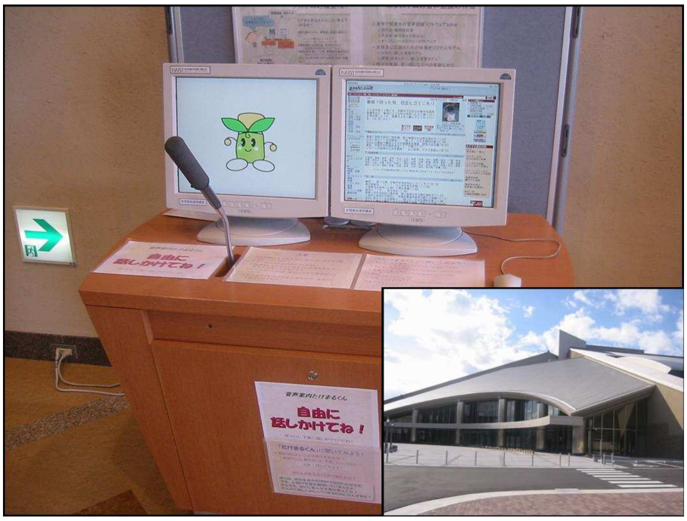

Figure 3.4. Speech-oriented guidance system *Takemaru*.

3.2 Speech-Oriented Guidance Systems

3.2.1 Purpose and Architecture

The purpose of a speech-oriented guidance system is to offer a certain group of users convenient access to proper information in a certain environment. While the information society is at the verge to an ubiquitous society, there is growing demand for this kind of services in any place. Although entering search queries via keyboard is still the prevailing method for accessing information, formulating one's question freely in natural language and using speech is a far more natural way to human-machine communication.

Figure 3.3 shows a block diagram of the main components of a speech-oriented guidance system. User input is recorded via a directivity microphone. After voice activity detection and rejection of non-verbal inputs, speech input is recognized in parallel using the open-source LVCSR engine Julius [37] with an adult and a child acoustic (AM) and language model (LM), respectively.

After age group classification, response generation is carried out. There is one question and answer databases (QADB) per age group. Each QADB contains a large number of question and answer pairs to cope with the wide variety of user questions. The response sentence corresponding to the example question most similar to the recognition result is selected.

Besides voice-based response message output, each system uses an extra screen to display a computer graphics agent and to display web pages. The presence of the agent gives the human user a virtual opponent to talk to in order to realize a more lively and natural human-machine interaction. The purpose of displaying web pages from the Internet is to give the user complementary information to the voice-based response.

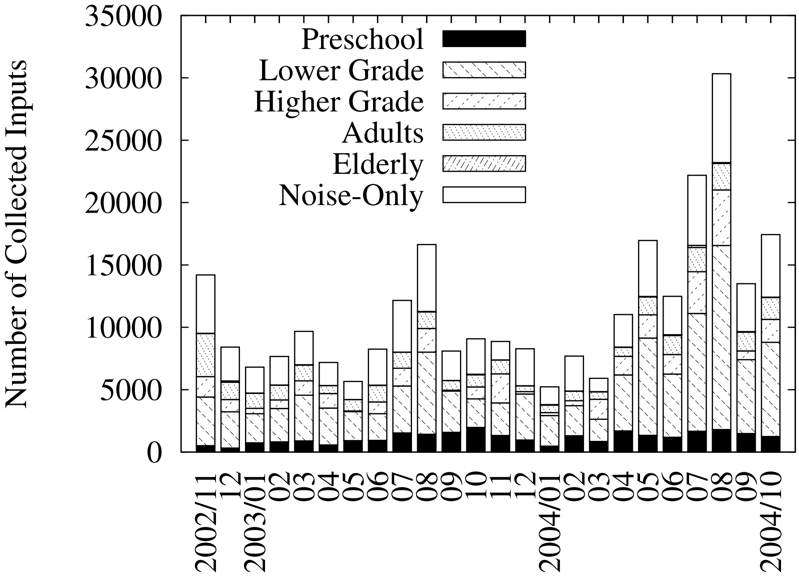

Figure 3.5. Inputs collected with *Takemaru* during two years of operation.

3.2.2 Takemaru

Takemaru is installed inside the entrance hall of the North community center in Ikoma city, Nara Prefecture, Japan since November 2002 (Figure 3.4). The indoor environment is relatively calm with a background noise level of approx. 50 dB(A). The place is frequently visited by adults and children, because it is a public facility with a library, a branch office for residental services and there are weekly events. The system uses the mascot character of Ikoma city, *Takemaru*, as agent. The *Takemaru* system can handle queries related to the agent, general information such as time, date, weather and news, the facility itself, surrounding area and sightseeing.

Takemaru has been collecting data for almost five years. The data of the first two years (2002/11 - 2004/10) are completely transcribed, labeled with tags (e.g. noisy, incomplete, invalid) and classified subjectively into five speaker groups (preschool children, elementary school children, junior-high school children, adults and elderly persons) by humans. Furthermore, utterances forming valid queries to the system, have been labeled with one or more possible system responses.

Figure 3.5 shows the number and age group classification of inputs collected during the first two years. Local peaks in the number of inputs are reached during the summer holidays in August 2003 and August 2004. Most of the inputs are from children showing that the employment of a speech-oriented guidance system such as *Takemaru* is a good way to collect spontaneous children speech.

More details about system architecture, adult/child discrimination, rejection of non-speech input (accuracy $\geq$ 85%) and preliminary results for recognition accuracy have been reported in [54] and [55].

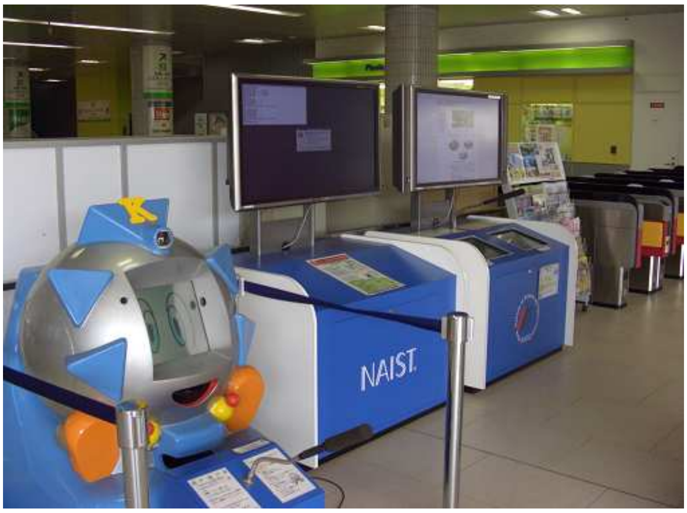

Figure 3.6. Speech-oriented guidance systems *Kita-robo* and *Kita-chan*.

3.2.3 Kita-chan and Kita-robo

The *Kita* systems are installed near the passenger gate of a subway station since March 2006. There is *Kita-chan*, a terminal-based system similar to Takemaru, and *Kita-robo*, a robot with moving eyes (Figure 3.6). Since the microphone of *Kita-robo* is installed at a relatively low position, more inputs of preschool children could be observed. On the other hand, the relative share of adult users is higher in case of *Kita-chan*. This might also be due to the system's outer appearance.

The agent's character and the robot's appearance are an imitation of the mascot of the subway station itself. No difference between the *Kita* systems will be made for system reuse and portability investigations. Although there is a roof above both systems, the environment is partly open-air. This is the main reason for a background noise level of approx. 60 dB(A), about 10 dB(A) higher than for the *Takemaru* environment. Fortunately, this is less problematic, because a directivity microphone is employed for sensing speech input. The contents of the *Kita* systems are an extension of the Takemaru system. They can also handle train information queries and display maps of certain areas or show the location of places of interest around the station, e.g. restaurants, shops, post offices, etc.

The *Kita* systems have been collecting data for almost two years. Inputs collected during the first nine months (2006/04 - 2007/01) of operation are transcribed and labeled by humans (Figure 3.7). Automatically detected noise inputs have been discarded in advance and were not transcribed. Moreover, only valid, human-transcribed user utterances from seven months (2006/04 - 2006/10, 2007/01) have been labeled with system responses.

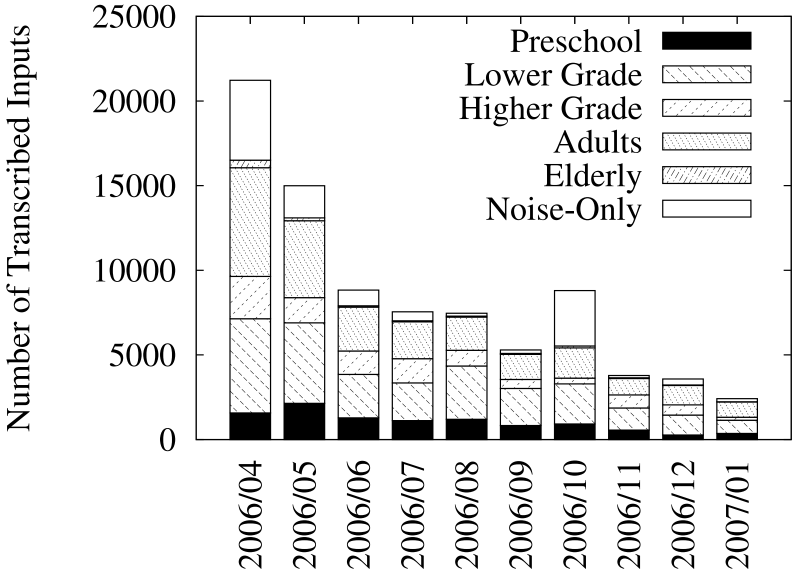

Figure 3.7. Human-transcribed inputs collected with *Kita-chan* and *Kita-robo* during the first nine months of operation.

3.2.4 Real-Environment Speech Database

A statistic of the number of inputs collected by end of December 2007 with *Takemaru* and *Kita* systems is shown in Table 3.1. When taking all systems together, more than 1.2 million inputs or more than 600 hours of real-environment speech and noise data have been collected since operation begin. There are more than 270,000 speech and noise inputs or 120 hours of human-transcribed data from *Takemaru* and more than 80,000 inputs or 40 hours from the *Kita* systems.

3.3 Development of Prototype System

Using two years of human-labeled real-environment speech data from *Takemaru* a development simulation is conducted in Section 3.3. The relationship between the amount of training data and various performance indicators is analyzed. Furthermore, the relative importance of developing each system component, i.e. acoustic model (AM), language model (LM) and question and answer database (QADB), is assessed.

The robust training of acoustic (AM) and language models (LM) for speech recognition requires a large amount of human-transcribed real speech data, since they are statistical models with a large number of parameters. For example, the PTM AM employed has more than 500,000 parameters and an N-gram LM may have up to V^N parameters for a vocabulary size of V words. Furthermore, the example-based response generation strategy will work the better, the more human-labeled question and answer pairs are available.

Table 3.1. Speech data collected with speech-oriented guidance systems *Takemaru*, *Kita-chan* and *Kita-robo* by end of December 2007

	Takemaru		*Kita*	
Classification	# Inputs	Time	# Inputs	Time
Transcribed	273,698	121.2 h	82,845	41.1 h
Preschool Children	27,535	14.3 h	10,115	5.4 h
Lower Grade	106,797	57.7 h	25,563	14.2 h
Higher Grade	31,402	15.8 h	9,980	4.8 h
Adults, Elderly	31,100	14.1 h	24,835	10.8 h
Noise, Non-Verbals	76,864	19.3 h	12,352	5.1 h
Untranscribed	684,461	334.6 h	186,830	107.6 h
Total	958,159	455.8 h	269,675	148.7 h

Table 3.2. Data employed for developing the *Takemaru* ASR module

Takemaru	Collection	Adult		Child	
Data Sets	Period	# Utter	Time	# Utter	Time
Training	22 months	16,332	8.2 h	75,315	41.4 h
Validation	1 months	3,069	1.5 h	4,115	2.3 h
Evaluation	1 months	1,085	0.5 h	6,568	3.7 h

However, for practical system development a trade-off between the costs for data preparation and the system's performance has to be made. Consequently, it is investigated how speech recognition and response accuracy evolve with increasing amounts of collected and human-transcribed data.

3.3.1 Experimental Setup

Table 3.2 shows the subset of the human-transcribed data employed for the long-term development simulation of *Takemaru*. Only valid user inputs have been labeled by humans with a correct system response and are employed in the simulation. Invalid inputs, i.e. meaningless, unintelligible, too noisy utterances, etc. were excluded, since they would not bring much benefit for constructing AM, LM or QADB. Data from November 2002 and August 2003 were put aside as validation and evaluation data, respectively.

ASR Module The amount of training utterances for each development period is given in Figure 3.8. Experimental conditions for AM training, LM training and speech recognition are given in Table 3.3. The LM for each training period is constructed by linear interpolation of the adult-dependent or child-dependent LM with the all data (adult and child data) LM. The interpolation weight was determined automatically so that the perplexity of the validation data set is

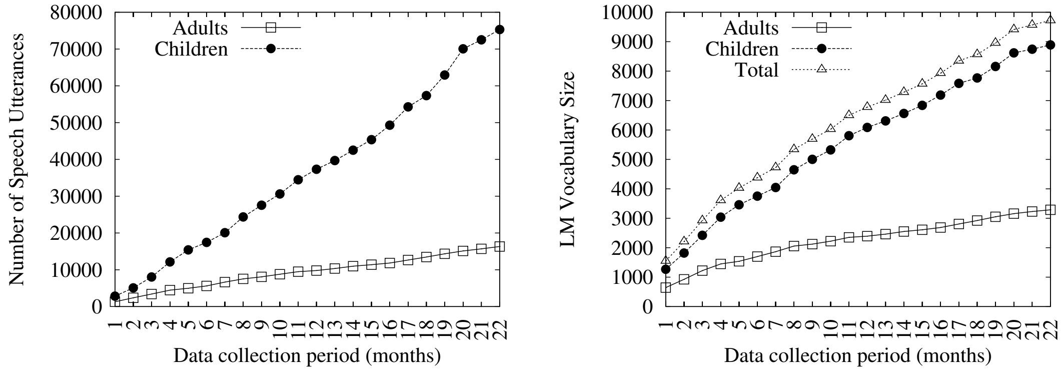

Figure 3.8. Left: Number of training utterances for each period. Right: Number of distinct words appearing in each training data set (Language model vocabulary size).

Table 3.3. Experimental conditions for *Takemaru* and *Kita* development simulation

AM Training	HTK 3.2 [31]
LM Training	SRILM 1.5.0 [72]
Acoustic Model	PTM [45], 2,000 states, 8,256 Gaussians
Acoustic Features	12 MFCC, 12 Δ MFCC, Δ E
AM Training	Baum-Welch, 3 Iterations
AM Adaptation	MLLR-MAP, 256 Classes, 3 Iterations
Language Model	3-gram, Kneser-Ney Smoothing
ASR Engine	Julius 3.5 [37]

minimized.

Since a user expects an immediate response from a dialogue system, speech recognition may not cause a delay before response generation is carried out. Consequently, a context-dependent, phonetic-tied mixture [45] acoustic model with relatively few parameters (8,192 Gaussians) is employed for real-time speech recognition. The Japanese Newspaper Article Sentences (JNAS) database [35] was employed to build the initial AM. This initial model is retrained with *Takemaru* speech data using either Baum-Welch training, or MLLR-MAP [48, 22] adaptation depending on the amount of available training data.

Q&A Module Each human-transcribed user utterance (= question, query) is labeled by humans with a correct system response (= answer). During the first months of operating *Takemaru*, new responses were added continuously if necessary to improve user satisfaction. The number of distinct Q&A pairs in the QADB for building the Q&A module is shown in Table 3.4. Pairs for utterances with a transcription appearing only once and which are linguistically unintelligible or out-of-domain were excluded from the QADB training data, because they had

Table 3.4. The number of distinct example questions and system responses in the question and answer database (QADB) as obtained from labeling the collected data. The total number of system responses (Max) is higher, because there were human-made responses never appearing in the collected data

QADB	Example Questions		System Responses		
System (Data Set)	Adult	Child	Adult	Child	Max
Takemaru (All)	6,671	32,992	275	285	322
Takemaru (Training)	4,052	17,891	265	282	322

a negative effect on response accuracy.

3.3.2 Performance Evaluation

Performance of a practical system will be limited by available technology and hardware resources. These limits are strict in practice, because development and production costs have to be kept low. On the other hand, there are imperative requirements such as real-time capability and a high level of user satisfaction. Taking standard ASR technology, standard hardware and real-time capability for granted, performance is considered as reasonable if additional training data does not improve performance significantly and it does not fall behind a comparable system, e.g. for automatic routing of telephone calls [26, 20, 27].

Vocabulary Size and Language Model Quality Figures 3.8 and 3.9 show the change in vocabulary size, out-of-vocabulary (OOV) rate and test set perplexity over time. The vocabulary size increases steadily over time. Almost ten thousand different words have been observed after 22 months and although the curve's slope becomes lower in the end, the vocabulary size would still increase largely beyond two years. This indicates that the domain of a speech-oriented guidance system is indeed open.

The OOV rate of the test data set decreases from about ten percent to a level of about one to two percent. This means that prediction of unknown words is still an important aspect to be dealt with in practice. The (Fix)-curves in Figure 3.9 show the test set perplexity when employing the two-year vocabulary for language model training, the (Var)-curves when using the monthly increasing vocabulary. The perplexity (word accuracy) does not decrease (increase) much after 12 months of data (47k sentences, 169k words) have been employed (Figures 3.8, 3.9).

Reliable estimation of n-gram probabilities usually require millions to billions of training sentences. However, it is difficult to obtain questions in written form about a local facility or subway station in large number from external language resources or web pages. Therefore, construction of a web-augmented language model is not considered in this work.

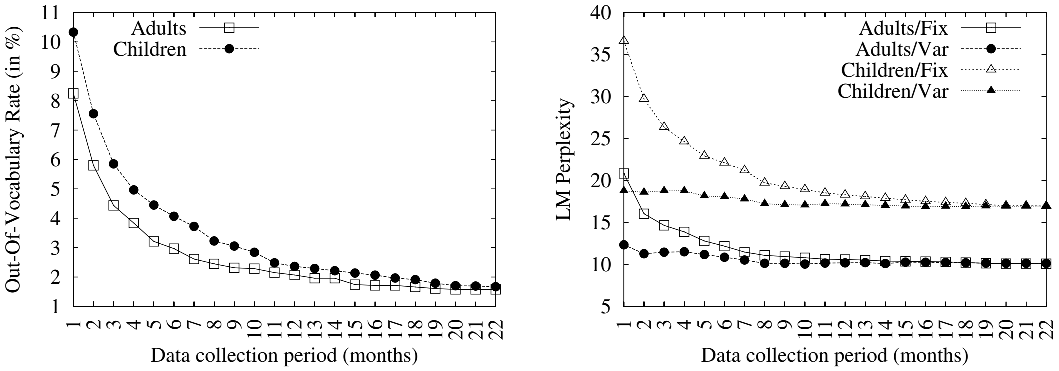

Figure 3.9. Left: Out-Of-Vocabulary (OOV) rate of each language model. Right: Test-set perplexity of each language model in case of a fixed two year vocabulary (Fix) and monthly increasing vocabulary (Var).

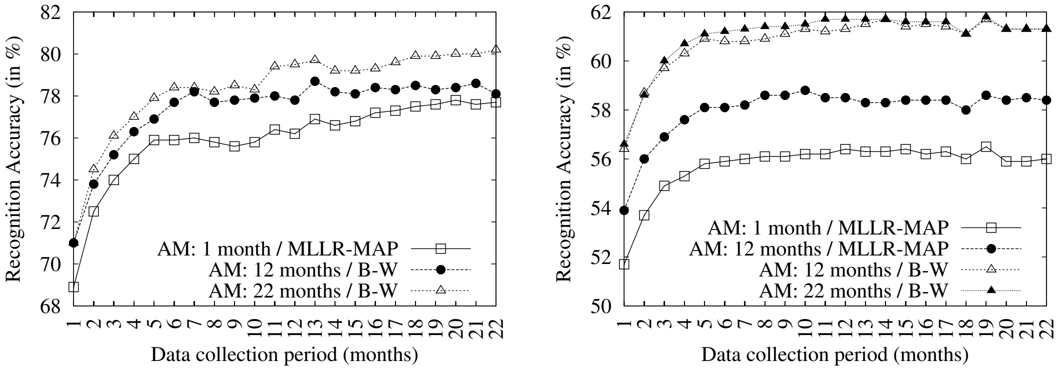

Figure 3.10. Effect of updating the language model (left: adults, right: children).

Speech Recognition Performance The relative increase in ASR performance due to language model update does not depend much on the update period of the acoustic model. There is a remarkable improvement with the data of the first six months (6k adult, 17k children utterances). However, there is only little or almost no improvement after the data of 12 months or more (10k adult, 37k children utterances) have been employed for language model training. This is in concordance with saturation of language model quality discussed before.

Figure 3.11 show the effect of updating the acoustic model. The language model is trained with all available data (22 months). The recognition accuracy for adult speakers is 74.3% when using the JNAS baseline model. Performance improves to 77.7% with only one month of adaptation data (1k utterances). There is not really a further improvement when using more data for adaptation. With Baum-Welch training there is a drop in performance in the beginning due to data insufficiency. Baum-Welch training outperforms model adaptation with more than 17 months (13k utterances) of data. A maximum in word accuracy of 80.2% is reached after 22 months (16k utterances). From the relatively moderate

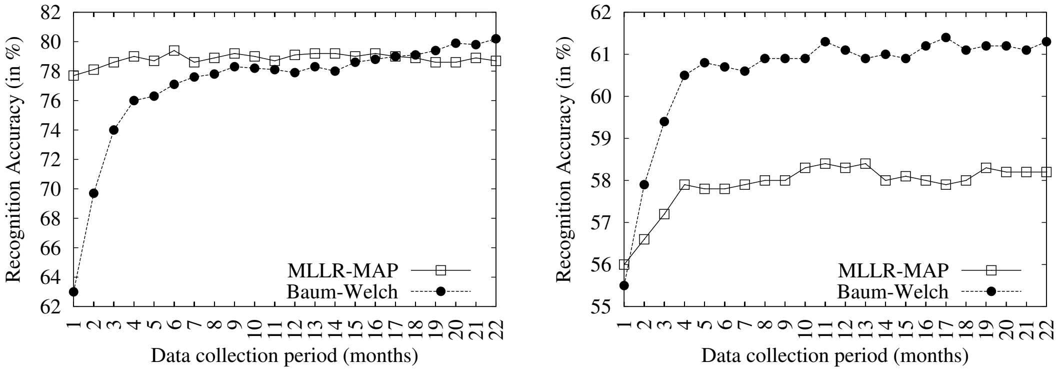

Figure 3.11. Effect of updating the acoustic model (left: adults, right: children).

improvement over the baseline it is clear that the acoustic differences between JNAS and *Takemaru* speech are small. This indicates that adult users seem to adapt and are willing to speak to the virtual agent *Takemaru* in a cooperative way.

A different tendency can be observed for children. The recognition accuracy with the JNAS baseline model is only 35.7%. This is due to the mismatch of JNAS data (adult, read speech) with *Takemaru* children data (child, spontaneous speech). There is a large improvement over the JNAS model when using only one to four months of speech data (3k to 12k utterances) for MLLR-MAP adaptation. Baum-Welch training outperforms adaptation already with two or more months of data (5k utterances or more). A performance of 60.5% is reached after four months (12k utterances). A peak of 61.3% was reached after 13 months (40k utterances).

The investigation shows that in order to train a speech recognizer with real-time capabilities about 10k to 15k valid user utterances will be required for reasonable performance. A maximum in ASR performance seems to be reaches with 40k-50k training utterances. The period necessary to collect this amount of data by operation in the target environment may depend on the speaker group.

Question and Answer Performance The considered application, speech-oriented guidance, is different from e.g. dictation in the sense that response accuracy is more important than plain recognition accuracy. Figure 3.12 shows the number of distinct question and answer pairs and distinct responses in the question and answer database (QADB) for each development period. Especially during the first four months the total number of different responses increases remarkably which is due to a permanent and active effort to add new responses during that period.

The effect of updating language model (LM), acoustic model (AM) and QADB is shown in Figure 3.13. An update of the QADB contributes most to an increase in response accuracy. Performance tends to saturate with about 3k adult and 13k children distinct Q&A pairs. In order to obtain 10k valid adult and 37k valid

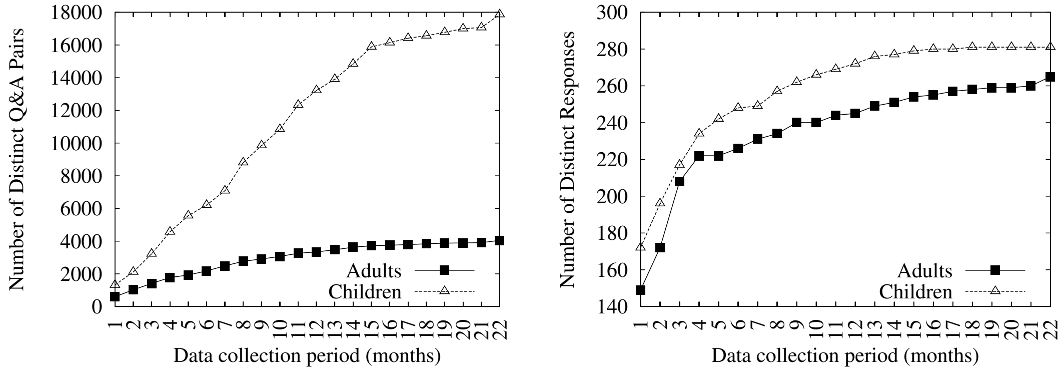

Figure 3.12. Number of distinct example queries and response sentences.

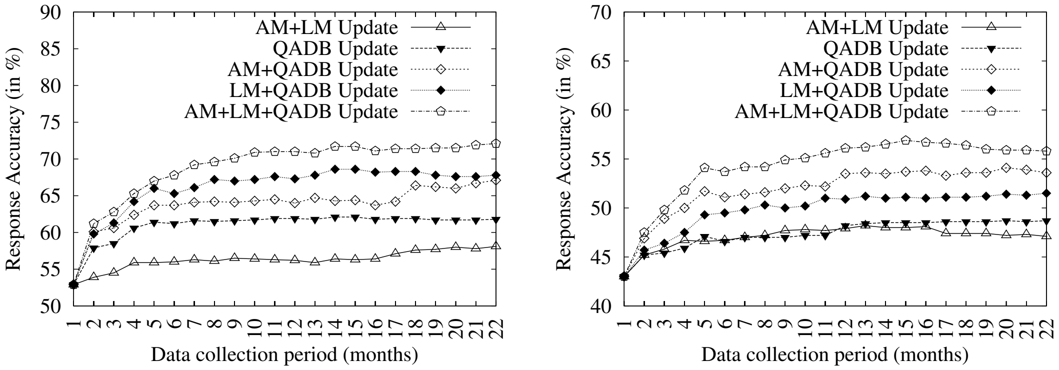

Figure 3.13. Response Accuracy (left: adults, right: children).

children utterances a collection period of 12 months was necessary.

From the results it is clear that response accuracy depends on AM and LM update. LM update is more important than AM update for adult users, because the initial AM was already constructed with adult speech so that LM update is relatively more effective. The opposite is observed for children users and is due to the same reason.

Finally the breakdown of response performance into subdomains is shown in Table 3.5. The performance for queries from adult users about weather, news, *Takemaru* agent, web access and city information is greater than 80%, and still greater than 70% for queries on the community center and local guidance. The performance for general interaction is only 40%. Nevertheless this is not problematic, since the corresponding queries are rarely information requests, but greetings, acknowledgments or expression of user emotions. The same is observed for children, where the response accuracy for general interaction was only 29%, which is the main reason for overall lower performance for children. Fortunately, most of the other information requests had a promising accuracy of about 70% and more.

Table 3.5. Subdomain response performance of the *Takemaru* system. Number of inputs and response accuracy (RA)

Speaker Group	Adult		Child	
Subdomain	# inputs	RA[%]	# inputs	RA[%]
General Interaction	**210**	40.0	**2179**	29.3
Agent, Takemaru	**492**	**81.7**	**2969**	**70.5**
Facility Guidance	112	69.6	509	**64.8**
Local Guidance	53	**73.6**	91	50.5
Date, Weather, News	155	**83.9**	623	**77.2**
City Information	38	**84.2**	132	39.4
Web Access	18	**83.3**	47	40.4
Other	7	28.6	18	22.2
Total	1085	72.1	6568	55.8

3.3.3 Conclusion

We have conducted a development simulation for a speech-oriented guidance system. Since task and domain of an open-domain dialogue system are defined by actual users of the system, real speech data collected in the target environment are required for system development. The development simulation was conducted to empirically determine the amount of data necessary for building each system component. There are acoustic and language model for speech recognition and the question and answer database for response generation.

A general statement about training data requirements for developing an arbitrary ASR application with reasonable performance cannot be made, because it depends on too many factors such as hardware resources, model complexity, domain complexity, target language, acoustic conditions, etc. Nevertheless, it is likely that the tendencies are the same if system architectures are comparable and considerations are restricted to a certain class of applications.

The collection and transcription of approx. 40k-50k high-quality utterances from adult and children users appear to be necessary until system performance can be considered as optimized. It took about one year to collect these data. However, it is in general impractical to collect this large amount of data over a longer period whenever a new system is to be developed. Reusing components of an existing system is a possible way to reduce development costs. Therefore, the portability of the community center guidance system *Takemaru* for the environment of the *Kita* systems, a local subway station, will be investigated in Section 3.4.

Since it is hard to find a comprehensive development report of a system similar to *Takemaru* in literature, comparison is restricted to a few selected studies. For example, improvement in recognition accuracy shows signs of stagnation after more than 40k in-domain sentences have been employed for language model training [1]. A further example is that the classification accuracy for up to 100

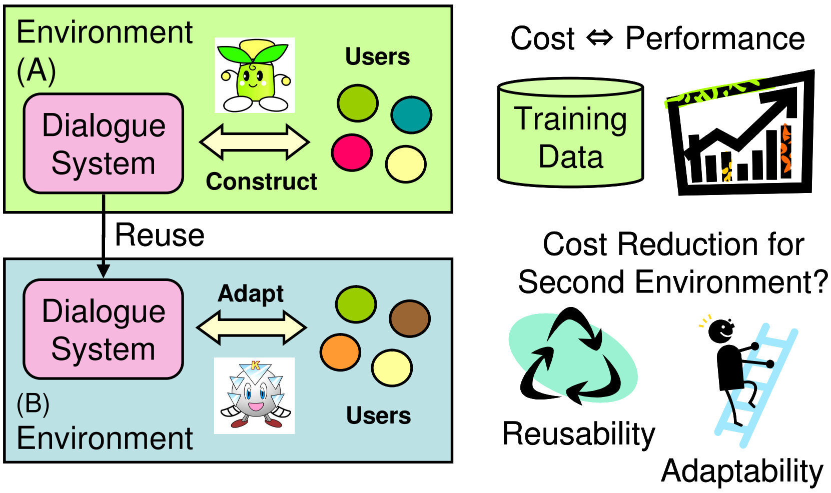

Figure 3.14. Reuse existing system in a different environment to investigate its portability and to identify possibilities for development cost reduction.

call-types does not improve after a comparable amount of training data has been employed [75].

The call routing application [26] is relatively suitable for comparison with the *Takemaru* guidance system. Employing approx. 10k and 30k labeled utterances for two different applications, a speech recognition and call classification accuracy of 70% and 74% are achieved, respectively [27]. In case of the *Takemaru* system a recognition accuracy of 80% and a response accuracy of 72% is achieved for adults users even with less than 20k task-specific training utterances. It is important to mention that there are two times more responses in the *Takemaru* system than call-types in the call routing application. Consequently, the performance of *Takemaru* can be considered as reasonable, because it does not fall behind a similar application.

3.4 Reuse of Prototype System

In Section 3.3 it has been shown that development of the real-environment speech-oriented guidance system *Takemaru* requires up to 40k-50k training utterances until speech recognition and Q&A performance saturates. It took about one year to collect these data. However, the high costs for collection (system installation in target environment, time for data accumulation) and preparation (human transcription and labeling) of this large amount of real-environment data would hinder any good business model for developing and selling such an application. Therefore, it is important to know in how far components of an existing prototype system can be reused and/or adapted to build a new system for a different target environment with reasonable performance but lower development costs.

In the following the portability of the speech-oriented guidance system *Take-*

Table 3.6. ASR and Q&A performance of *Takemaru* measured by word accuracy (WA) and response accuracy (RA)

Takemaru	# Utterances		Adult [%]		Child [%]	
Training Period	Adult	Child	WA	RA	WA	RA
1 month (short-term)	1k	3k	68.9	52.9	52.1	43.0
6 months (medium-term)	6k	17k	77.4	67.8	60.8	53.7
22 months (long-term)	16k	75k	79.5	72.1	62.0	55.8

maru is investigated. A system or a module of a system can be called portable if system performance is high before adaptation to a new target environment. It can also be considered as portable if performance improves remarkably with moderate amounts of adaptation data and performance improvement shows signs of stagnation. Consequently, the notion of portability comprises both reusability and adaptability

The initial prototype system *Takemaru* will be adapted to a different environment, a local subway station, in order to construct the *Kita* systems [39]. The effect of reusing ASR and Q&A component of the *Takemaru* system with and without adaptation using moderate amounts of real-environment data collected in the *Kita* environment is investigated. Furthermore, the level of performance in both environments is compared.

The investigation will show that it is possible to reduce the development costs of real-environment, speech-oriented guidance system by reusing data and models of an existing prototype system. Although the data collection period is shorter and the amount of human-transcribed speech data employed for adaptation is smaller, a speech recognition and response performance comparable to the *Takemaru* system are achieved.

3.4.1 Prototype System

Before investigating the portability of the *Takemaru* system, data requirements (amount of training data) and performance results (WA, RA and LM quality) for short-term, medium-term and long-term development of the initial *Takemaru* prototype system are reviewed.

System performance has been evaluated in case of short-term (one month, 4k data), medium-term (six months, 23k data) and long-term (22 months, 91k data) development. Word and response accuracy are given in Table 3.6. It is clear, that the performance improvement from short-term to medium-term development is quite large but relatively small from medium-term to long-term development. Children performance saturates earlier than adult performance because the number of available adult data is only small. Consequently, long-term development may be required for developing the initial prototype system in practice.

Table 3.7. Vocabulary Size [words], OOV rate [%] and test set perplexity (PP) of *Takemaru* LM

Takemaru Training Period	Adult LM Vocab	Adult LM OOV	Adult LM PP	Child LM Vocab	Child LM OOV	Child LM PP
1 month (short-term)	0.6k	8.2	20.7	1.2k	10.3	35.9
6 months (medium-term)	1.6k	3.0	12.0	3.6k	4.1	21.6
22 months (long-term)	3.1k	1.6	9.9	8.5k	1.7	16.5

Table 3.8. Training and evaluation data collected in *Kita* environment

Kita Development Speech Data Sets	Collection Period	Adult Data # Utter	Adult Data Time	Child Data # Utter	Child Data Time
Kita (Training)	6 months	11,276	5.5 h	18,720	10.5 h
Kita (Evaluation)	14 days	1,699	49 m	2,732	91 m

A similar tendency can be observed for vocabulary size and LM quality as given in Table 3.7. When comparing medium-term and long-term development, the vocabulary size more than doubles, the OOV rate is reduced by more than 50% and there is a significant reduction in perplexity.

3.4.2 Task Data for System Update

Speech utterances forming valid queries collected during one month (short-term development/adaptation period) to six months (medium-term development/adaptation) of *Kita* system operation are employed for updating AM, LM and QADB. 14 days of user inputs collected during the first half of May 2006 are employed for performance evaluation (Table 3.8).

ASR Module For AM adaptation to the acoustic environment and due to the comparably low amount of adaptation data, MLLR-MAP adaptation is employed. Baum-Welch retraining of the *Takemaru* AM using all available *Takemaru* and *Kita* training data could not outperform MLLR-MAP adaptation. The LM is constructed from the human transcriptions of all available *Takemaru* and *Kita* training utterances.

Q&A Module The initial Q&A DB for the *Kita* systems consists mainly of human-labeled Q&A pairs collected during the first months of operating *Takemaru*. They have partially been edited by humans for the *Kita* systems. The *Kita* QADB is updated with Q&A pairs obtained during one month to six months of operating the *Kita* systems. The number of distinct Q&A pairs in the QADB after adding the newly collected pairs almost triples. 75 new response sentences have also been added for user queries with no appropriate counterpart available

Table 3.9. Number of distinct example questions and system responses in the *Kita* QADB after medium-term update

Q&A Database System (State/Set)	# Questions Adult	 Child	# Responses Adult	 Child
Reuse *Takemaru*	2,761	5,062	183	179
Kita (Collected)	5,515	10,252	298	300
Kita (Training)	4,830	8,867	289	295
Kita (Updated)	7,018	13,022	315	320

Table 3.10. ASR and Q&A performance of the *Kita* systems measured by word accuracy (WA) and response accuracy (RA)

Development Strategy	# Utterances		Adult [%]		Child [%]	
(System Update Period)	Adult	Child	WA	RA	WA	RA
Reuse *Takemaru*	20k	86k	74.3	52.0	57.0	45.5
+ 1 month *Kita* (short-term)	+ 5k	+ 7k	77.0	67.2	59.9	54.7
+ 6 month *Kita* (medium-term)	+ 11k	+ 19k	78.7	70.8	60.4	58.1

in the existing response set (Table 3.9).

3.4.3 Experimental Results for System Update

The ASR and Q&A performance before and after system update is given in Table 3.10. With the short-term (one month, 12k data) update the absolute improvement in word accuracy is only moderate (adult: 2.7%, child: 2.9%). The medium-term (six months, 30k data) update yields only very small additional improvements (adult: 1.7%, child: 0.5%) over the short-term update. The absolute improvement for adults is slightly higher than for children. This is likely to be due to the fact that more child data has been available for constructing the models of the *Takemaru* ASR module. The quality for initial and updated LMs is shown in Table 3.11. There is only a small increase in vocabulary size and small decrease in perplexity. This shows that the *Takemaru* ASR module has a high portability in the *Kita* environment.

Table 3.11. Vocabulary size [words], OOV rate [%] and test set perplexity (PP) of the LM for the *Kita* systems

Development Strategy	Adult LM			Child LM		
(System Update Period)	Vocab	OOV	PP	Vocab	OOV	PP
Reuse *Takemaru*	3.7k	2.9	17.5	10.3k	2.4	25.8
+ 1 month *Kita* (short-term)	4.3k	1.8	15.9	10.7k	1.9	25.1
+ 6 month *Kita* (medium-term)	4.8k	1.5	14.5	11.2k	1.7	23.9

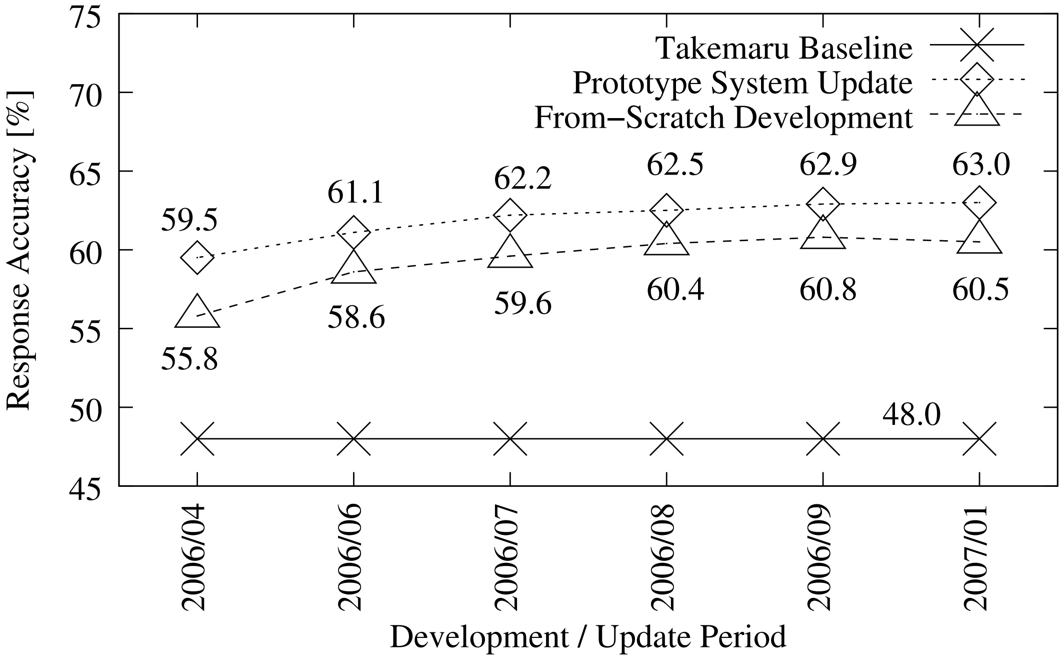

Figure 3.15. Response accuracy for *Kita* data of *Takemaru* baseline system, from-scratch development and updated *Takemaru* system.

Considering simultaneous adaptation of ASR and Q&A modules, there are remarkable improvements in response accuracy after short-term (15.2% and 9.2%) and further significant improvements after medium-term (3.6% and 3.4%) update. Although performance is low with the initial, human-edited QADB, response accuracy rebounds after short-term update. A level comparable to the *Takemaru* system is reached after medium-term update. This indicates medium portability of the *Takemaru* Q&A module in the *Kita* environment.

3.4.4 System Update vs. From-Scratch Development

The results from the last section do not yet show whether performance improves by reusing an existing system. Therefore, development by reusing and updating the components of the *Takemaru* system is compared with from-scratch development.

Figure 3.15 compares the performance of system reuse without update, after system update and from-scratch development. There is an absolute performance improvement of 2.1% - 3.7% in response accuracy (adults and children combined) over from-scratch development by reusing AM, LM and QADB of *Takemaru*. This shows that reusing an existing prototype can be successful to boost performance.

Furthermore, it has to be investigated whether development costs can be reduced by system reuse. Table 3.12 compares the costs of from-scratch development with prototype system update for two selected developments periods with almost equal results for Q&A performance. It is clear that the development

Table 3.12. Cost performance of system reuse vs. from-scratch development

Development		Performance		# Data	Costs	
Strategy	Period	RA	WA	Employed	[points]	Reduct.
System Reuse	1 month	59.5%	66.1%	16,525	513,125	42%
From-Scratch	3 months	59.6%	63.8%	31,489	887,225	-
System Reuse	2 month	61.1%	66.4%	24,466	711,650	41%
From-Scratch	5 months	60.8%	63.8%	43,887	1,197,175	-

period can be reduced by more than half (3 → 1 month, 5 → 2 months) and that data collection and transcription costs (31k → 17k data, 44k → 24k data) can be reduced by more than 40% without compromising the performance. It is worth mentioning that word accuracy is about 2.5% higher in case of system reuse although response accuracy is about the same.

3.4.5 Takemaru Development vs. Kita System Update

The results of the *Takemaru* development simulation and update simulation for the *Kita* environment are summarized in Figure 3.16. They show the relationship between development period (amount of training data) and ASR and Q&A performance for from-scratch development and system reuse and update. Furthermore, it is possible to compare the level of performance of *Takemaru* with *Kita-chan* and *Kita-robo*.

Since the perplexity of the *Kita* evaluation data is higher than that of the *Takemaru* data, ASR performance of the *Kita* systems can be considered as reasonable if it is equal to or higher than that of *Takemaru*. From Figure 3.16 it is clear that the ASR performance of the long-term *Takemaru* ASR module is higher in the *Kita* environment than ASR performance of the short-term *Takemaru* module in the *Takemaru* environment. The same can be observed for medium-term development of the *Takemaru* ASR module and short-term adaptation of the *Takemaru* module for the *Kita* environment. However, the improvement of medium-term adaptation over short-term adaptation is small.

This indicates that reusing the *Takemaru* ASR module for the *Kita* environment can reduce the development period from medium-term (six months) from-scratch development to short-term adaptation (one month) while maintaining the same level of ASR performance. Furthermore, it seems that medium-term to long-term adaptation of the ASR module can be avoided.

Reuse of the *Takemaru* Q&A module for the *Kita* environment is less effective than a reuse of the ASR module. Although the same level of response accuracy is reached when comparing the performance of *Takemaru* system reuse in *Kita* environment and *Takemaru* short-term from-scratch development, and short-term adaptation and medium-term from-scratch development, the relative improvement of response accuracy is much higher than the relative improvement of word

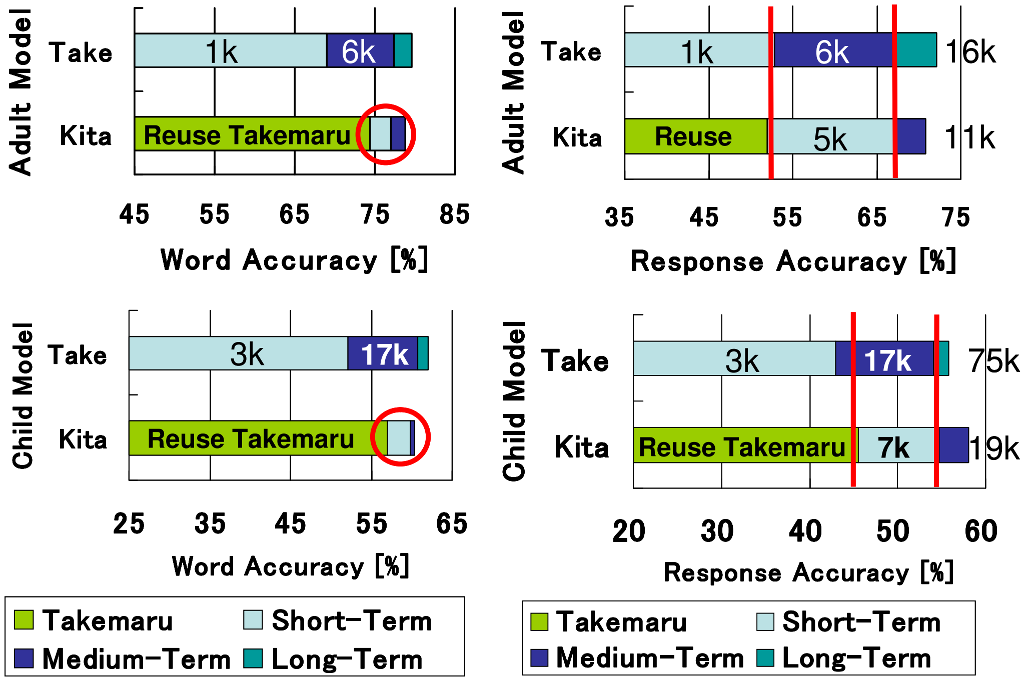

Figure 3.16. Result of portability experiment for *Takemaru* → *Kita* environment. The reusability of the ASR module is high and performance is almost saturated with short-term adaptation. However, portability of the Q&A module is only medium, because there are remarkable improvements after shor-term and medium-term adaptation.

accuracy for the corresponding adaptation periods. Although it seems possible to avoid long-term adaptation of the Q&A module, medium-term adaptation is at least recommendable.

For adults, the preparation of labeled adaptation data to achieve the same level of performance can be reduced by more than half (17k → 7k) case of system reuse in comparison to from-scratch development. For children, a reduction of more than two thirds (75k → 19k data) could be achieved. The higher portability of the childrens' models seems to be due to the fact that more training data from children have been available for *Takemaru*.

3.5 Domain Analysis and Comparison

Although the portability of the Q&A module has been assessed using the response accuracy after system update, an approach which gives more insight would be preferable. Therefore, the domain difference between the *Takemaru* and the *Kita* environment is assessed by directly comparing the QADB contents and domain-specific language models. Several methods to measure the domain difference and degree of portability will be investigated. Furthermore, a subdomain analysis is conducted. Finally, the most frequent user queries and keywords for the intersection and each system-specific domain are determined.

3.5.1 Domain Distance

The domains of the *Takemaru* (A) and *Kita* systems (B) are compared using human-labeled Q&A pairs and language models trained on transcriptions of user utterances. In order to determine that part of the *Takemaru* QADB which is reusable for the *Kita* systems, a mapping of corresponding system responses was established between the two systems.

Let a_i and b_i be the relative frequency of system response i in domain A and B, respectively. $\overline{a}$ and $\overline{b}$ denote the average response frequency, x denotes any n-gram. Three measures for domain distance can be defined as follows:

- Correlation between the response frequencies a_i and b_i

$$COR(A,B) = \frac{\sum\limits_i (a_i - \overline{a})(b_i - \overline{b})}{\sqrt{\sum\limits_i (a_i - \overline{a})^2 \sum\limits_i (b_i - \overline{b})^2}}$$

- Probability of the response intersection set of both systems, counting repeated occurrences of responses for all utterances and users

$$P(A \cap B) = \sum_i \min\{a_i, b_i\}$$

- Symmetric KL distance (KLD) between the n-gram $P_A(x)$ and $P_B(x)$ of domain-specific language models

$$\frac{1}{2}\left[D(P_A||P_B) + D(P_B||P_A)\right] = \frac{1}{2}\sum_x \left[P_A(x)\log\frac{P_A(x)}{P_B(x)} + P_B(x)\log\frac{P_B(x)}{P_A(x)}\right]$$

There are responses i which are specific to either domain (A) or (B). Therefore, response frequencies of zero were smoothed using the Good-Turing method. Two domains are the more similar the higher the value of COR and P(A∩B) are, and the lower the value of KLD is.

Figure 3.17 shows the cross-domain histogram of relative response frequencies. A certain degree of similarity between the *Takemaru* and *Kita* domain is obvious from the response distribution. The domain distance using the objective measures correlation and intersection probability are given in Table 3.13. The distance between two random subsets of the *Takemaru* data is also shown as reference. Since there is a very high correlation (1.00) and a high probability for the intersection set A∩B (0.93) the proposed metric can be considered as valid.

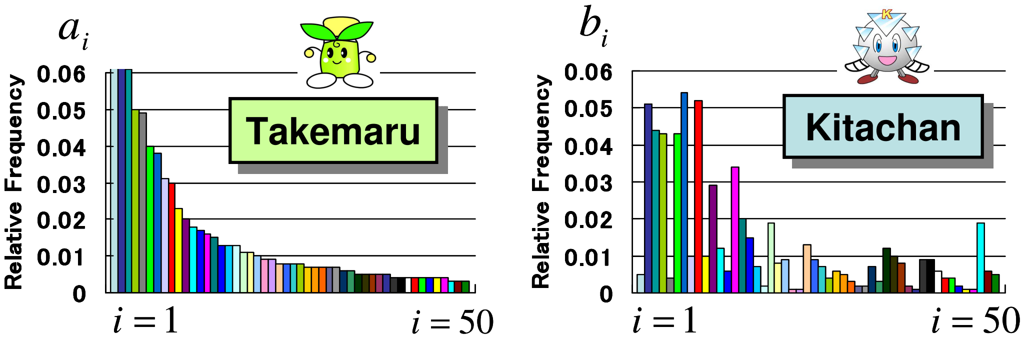

Figure 3.17. Cross-domain histogram of relative response frequencies.

Table 3.13. Domain comparison using response statistics

Responses A ↔ B	COR	A∩B
Takemaru ↔ Takemaru (Subsets)	1.00	0.93
Takemaru ↔ Kita (All)	0.59	0.56

When comparing *Takemaru* and *Kita* domain, a value of 0.56 for P(A∩B) indicates that at least half of the users' inputs to *Takemaru* have also been observed for the *Kita* systems and vice versa. The correlation has a similar value of 0.59.

These results indicate that objective domain distance measures such as the correlation or the intersection set probability can be employed to assess portability. Correlation and intersection set probability are normed between 0 and 1. Therefore, COR(A,B) of 0.56 and P(A∩B) of 0.59 are in concordance with the notion of 'medium' portability of the *Takemaru* Q&A database in the *Kita* environment as it has been assessed in the previous section using the response accuracy.

The number of words in the intersection and union of domain vocabularies as well as the KL divergence between domain-specific uni-gram language model probabilities are shown in Table 3.14. It is clear that more than half of the words from the *Kita* domain also occur in the *Takemaru* domain. Combining the words of both domains there is a moderate increase (11%) of vocabulary size in comparison to the *Takemaru* domain. Moreover, 74% of *Kita* words have also been observed in the *Takemaru* domain.

Table 3.14. Domain comparison using language models

Data	*Takemaru* (A)	*Kita* (B)	A∪B	A∩B	LM-KLD
Set	# words	# words	# words	# words	[bit]
All	11,192	4,768	12,430	3,530	8.57
Adult	3,782	2,625	4,865	1,542	0.52
Child	10,344	3,696	11,172	2,868	10.13

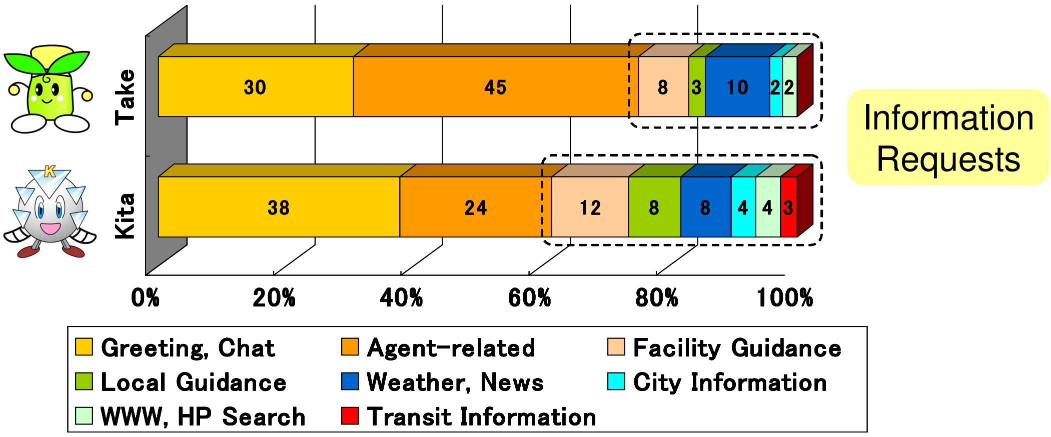

Figure 3.18. Subdomain analysis and comparison.

Although the KLD is difficult to interpret, it is of practical interest if more than two domains are compared. Imagine that several databases for different domains are available. Moreover, it is known that the target domain of a new system to be developed is similar to one of them. Then it is worth considering to reuse the data also from close domains, i.e. the inter-domain KLD is small.

3.5.2 User Utterance Contents

The domain comparison did not reveal the users' utterances contents regarding the common and environment-specific domains. It is possible to group user utterances by the response sentence indices, which have been assigned by human annotators. Furthermore, it is possible to group response sentence and corresponding input sentence into subdomains.

The result of a subdomain analysis for valid user inputs is given in Figure 3.18. It is clear that more than half of the users' utterances are greetings (e.g. hello and good-bye), chat (e.g. how do you do? smart! cute! stupid!) and agent-related questions (e.g. what is your name? what are your hobbies?). The remaining user inputs are information requests about the facility, local area, weather, news and webpage access. These information requests are the system domain originally intended by the inventors. It is surprising that only three percent of the valid user inputs are transit information requests, although the *Kita* systems are installed at a train station.

The response accuracy of each subdomain for the evaluation data is shown in Table 3.15. There is a response accuracy of 79% or more for adult queries related to the agent, facility guidance, weather, news and time information. The accuracy for local guidance, city and transit information is above 60%. Although overall response accuracy for children is only 58%, 65-86% of childrens' queries related to the agent, local guidance, weather, news and time information are processed correctly.

Table 3.15. Subdomain response performance of the *Kita* system. Number of inputs and response accuracy (RA)

Speaker Group	Adult		Child	
Subdomain	# inputs	RA[%]	# inputs	RA[%]
General Interaction	**458**	62.7	**1316**	53.0
Agent-related	**320**	**80.9**	**651**	**70.4**
Facility Guidance	195	**79.0**	263	57.0
Local Guidance	243	68.7	142	**65.5**
Weather, News, Time	148	**94.6**	116	**86.2**
City Information	92	65.2	83	54.2
WWW, HP Search	155	51.6	80	17.5
Transit Information	73	68.5	50	22.0
Other	15	40.0	31	64.5
Total	1699	70.8	2732	58.1

Finally, an insight into actual data is given. The ten most frequent system responses of the intersection domain, i.e. responses x in the intersection set with highest probability $P_A(x)P_B(x)$ are shown in Table 3.16. Most of them are greetings, agent-related information requests and weather forecast.

A ranklist of the relatively most frequent responses in domain (A) or (B) can be obtained by listing the responses by the weighted probability ratio $P_A(x)\log[P_A(x)/P_B(x)]$ or $P_B(x)\log[P_B(x)/P_A(x)]$ in descending order, respectively. The list of relatively frequent *Takemaru* and *Kita* responses is given in Tables 3.17 and 3.18, respectively.

While *Takemaru* users are often concerned about current time, bus timeable and local information, *Kita* users are mainly interested in the local map, location of restaurants, post office, etc.

A similar list can be obtained for keywords in user utterances using uni-gram language model probabilities (cf. Table 3.19). It is interesting to see that among the keywords for the *Takemaru* domain are objects related to the environment (room, library, book), the agent's name and that it looks 'cute'. For the *Kita* domain there is also the agent's name, environment-related objects (station, vending machine), places near to the station (restaurant, SanMarc, NAIST, Kitayamato, Mayumi) and farer locations (Kyoto, Namba).

3.5.3 Related Work

Related work, e.g. for call routing [20] shows that data from previous developments of ASR applications are often effective to bootstrap a new application. The ASR performance gap between from-scratch development with in-domain data and reuse of an existing prototype system was only about five percent. At the same time a reduction of the OOV rate was achieved. Furthermore, the ex-

Table 3.16. Responses very common in *Takemaru* and *Kita* domain

Response Meaning	Response Sentence
(greeting)	“Hello.”
(weather forecast)	“Tomorrow’s weather will be ...”
(agent, self-intro)	“My name is ...”
(place, toilet)	“The toilet is ..”
(greeting)	“See you again.”
(agent, current age)	“I am ... year(s) old.”
(agent, favorite food)	“My most favorite food is ..”
(websearch)	“Please tell me the search keyword.”
(greeting)	“You’re welcome. Please come again.”
(newspaper)	“I show you the newspaper page.”

Table 3.17. Responses relatively frequent for *Takemaru* domain

	Response Meaning	Response Sentence
0.25	(out-of-domain)	“I am sorry, but I do not know.”
0.12	(current time)	“The time is ...”
0.07	(greeting)	“Hello.”
0.05	(agent, self-intro)	“My name is ...”
0.03	(bus information)	“I show you the bus timetable.”
0.03	(agent, appearance)	“Don’t you think I am cute?”
0.02	(offer friendship)	“Please become my friend.”
0.02	(misunderstanding)	“I do not understand you.”
0.02	(local information)	“You are at the community center.”
0.01	(offer information)	“Please ask me something about ...”

Table 3.18. Responses relatively frequent for *Kita* domain

	Response Meaning	Response Sentence
0.14	(map, local)	“I show you the local map.”
0.05	(map, restaurant)	“I show you the restaurant map.”
0.04	(place, toilet)	“The toilet is ...”
0.03	(map, post office)	“The nearest post office is ...”
0.03	(agent, origin)	“My name is Kita because ...”
0.03	(weather forecast)	“Tomorrow’s weather will be ..”
0.02	(user warning)	“Please do not tease me.”
0.02	(misunderstanding)	“Could you please say that again?”
0.02	(general response)	“How may I help you?”
0.02	(bus information)	“Please take the south exit.”

Table 3.19. Keywords in user utterances with a high probability for both domains and which are relatively more frequent in one domain than the other domain

Common Domain	where? Takemaru Ikoma what? who? here like news station name today now you toilet stupid when? search weather forecast tomorrow born goodbye cute understand say Mayumi how_old?
Takemaru Domain	Takemaru begin search understand now what? birthday friend when? you stupid pool game live kindergarten bus cute home what? room sleep library front who? noisy sing book
Kita Domain	Kita today vending machine line where? news Kyoto like map restaurant Ikoma Kuragaritoge show weather SanMarc NAIST Kitayamato nearby go tell_me Namba Nara station Mayumi

isting system covered already about 70% of the new application's call types. The effect of combining in-domain with existing data was not investigated in the same study. The result of a later study about multi-task learning for the same task was that reusing labeled utterances available from previously developed applications do not improve the performance of call-type classification [76]. This indicates that data reuse must not always be successful.

3.6 Conclusion

For the same reasons as outlined in Section 3.3.3 it is difficult to make general statements regarding system portability. In this chapter we have considered the special case of the *Takemaru* and *Kita* speech-oriented guidance systems. Although both systems share the same architecture, they have been installed in different environments and exposed to different users. The portability of the *Takemaru* system has been investigated by conducting a development simulation for the prototype system and update simulation for the *Kita* environment. The purpose of the investigation is to assess whether reusing the *Takemaru* ASR and Q&A module in the *Kita* environment is effective for reducing the development costs. Since it is impractical to collect and prepare large amounts of real-environment speech data over a long time span, it is imperative to shorten the development cycle and to reduce adaptation data requirements.

The investigation in Section 3.4.4 showed that it is possible to reduce development costs for data collection and transcription by more than 40% when reusing the *Takemaru* prototype for development of the *Kita* systems. At the same time the development period is reduced by more than half (3 → 1 month, 5 → 2 months) without compromising the performance.

Results of the portability investigation in Section 3.4.5 show that the *Take-*

maru ASR module has a high degree of reusability. Performance improvement stagnates already with small amounts of adaptation data from the new target environment. On the other hand, the portability of the *Takemaru* Q&A module in the *Kita* environment was only medium. Therefore, medium-term adaptation of the Q&A module is recommendable. The comparison of *Takemaru* development and *Kita* update revealed that the same level of ASR and Q&A performance can be reached after a shorter adaptation period of the *Kita* systems by reusing *Takemaru*.

Finally, Q&A portability was also assessed by measuring the cross-domain correlation of relative response frequency, intersection set probability and KL divergence of N-gram probabilities.

Chapter 4

Selective Training for Task-Adaptation

Statistical models like HMMs employed for acoustic modeling in speech recognition have a large number of parameters. To reliably train such models a huge speech database is required. However, the collection of speech data, including recording and transcription, is a very time-consuming and costly process (cf. Section 1.3.2). Moreover, there is the task-dependency of speech recognition, i.e. the acoustic model has to cope with many speech variabilities, e.g. speaker characteristics, speaking style, acoustic conditions and so on (cf. Section 1.3.1). Consequently, it is impractical to provide enough training data for each possible combination.

Therefore, our goal is the development of a method which enables the construction of task-adapted acoustic models automatically and without much or no additional costs for collection of task-specific data. One idea of the approach is to reuse existing speech data. The idea of multi-source training as introduced in Section 1.5.2 is to combine several existing databases to construct generic acoustic models to build a task-independent ASR system. The reuse of speech data and models of a speech-oriented guidance system for a different environment has also been investigated in Chapter 3.

However, instead of just reusing all available speech data for training, the idea of the proposed selective training framework is to employ only a subset of all available speech data to build a task-adapted acoustic model. This requires a method to select those user utterances from a large data pool which are close to the desired target task. Although we are focusing only on selective training of HMM-based acoustic models, the proposed method is applicable to any statistical model which has sufficient statistics.

4.1 Conventional Data Selection Methods

In recent years, proposals for training procedures which make selective use of training data, emerged in literature. A selective training method for HMMs is described in [4]. Each training sample is weighted by a confidence measure in order to control the influence of outliers. The approach was applied to improve the statistical models for accent and language identification.

Active learning (cf. Section 1.4.2) is employed in [38] in order to reduce the effort necessary for database preparation. Only those utterances with a low recognition confidence score are transcribed and employed for training. Collection of additional data is only carried out as long as the likelihood of the trained model given the selected training data is no longer increasing. Experiments revealed, that the best model is not necessarily obtained when using the whole database for training, but when only using a subset of the whole data.

There are adaptation methods which make selective use of data. Training speakers which are close to the test speaker are chosen based on the likelihood of speaker GMMs given the adaptation data [82, 25]. The adapted model is constructed from combining precomputed HMM sufficient statistics for the training data of the selected speakers. A similar paradigm is employed in [32], where cohort models close to the test speaker are selected, transformed and combined linearly.

In order to select speech data from a large data pool, the selection procedure from [82, 32, 25] is not applicable, if the speaker label of each utterance is unknown or if there are only few utterances per speaker. This is the case for data automatically collected by a dialogue system for public use such as Takemaru-kun [53].

4.2 Proposed Data Selection Method

In the following an framework for building task-adapted models is described. Furthermore, an optimization criterion and a selection algorithm to implement the framework are proposed. The model likelihood given a small amount of development data is employed as optimization criterion. The development set has to be designed to represent the desired target task well. In order to make the approach computationally feasible, a greedy algorithm for selecting an appropriate data subset from a large pool of existing speech data has been developed.

The proposed optimization criterion is theoretically more well-defined in comparison to a heuristic weighting of frames like in [4] and the likelihood-based stopping and the margin-based selection criterion from [38]. The selection unit (utterance) is larger than in [4] (frame) but smaller than in [82, 25] (speaker).

4.2.1 Selective Training Framework

Consider the scenario illustrated by Figure 4.1. One or more rather large speech databases are available. The conglomerate of several databases is called *training data pool*. Our goal is to obtain an acoustic model for a certain speech recognition task, e.g. dictation, human-machine dialogue, human-human conversation, etc.

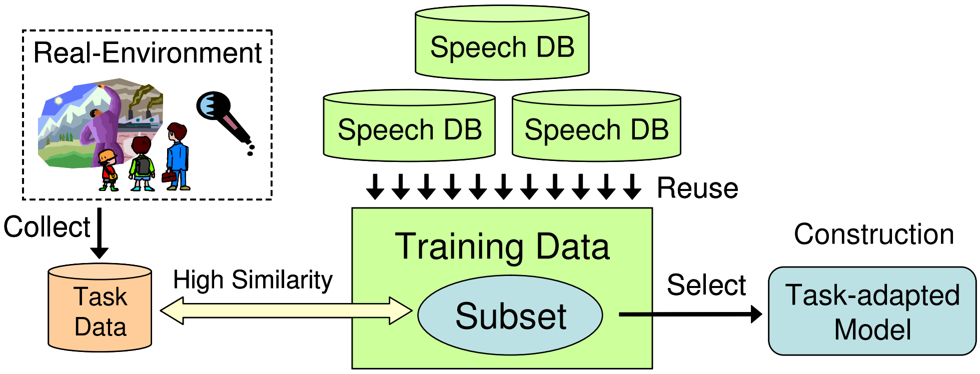

Figure 4.1. Proposed selective training framework.

However, there is only a small amount of task-specific data available. If there is no data, just a small development set has to be collected. The costs for providing this data would be far lower than the collection of a larger amount of task-specific training data. If there are not enough task-specific data for model training or adaptation, it is desired to augment the development data set by selecting utterances close to the development data from a large pool of existing speech data.

4.2.2 Optimization and Selection Criterion

The log-likelihood function $P(\mathcal{X}|\boldsymbol{\Theta})$ (Eq. 2.3) is a measure of how well the random sample $\mathcal{X}$ fits the model $f(\boldsymbol{x}|\boldsymbol{\Theta})$. The task of statistical model estimation, e.g. MLE (Section 2.2.1), is to find an estimate $\hat{\boldsymbol{\Theta}}$ for the model parameters which maximize the log-likelihood function using the given sample $\mathcal{X}$.

In case of the selective training framework, the data set for calculating the model parameters $\mathcal{T}$ and the data set for calculating the likelihood $\mathcal{D}$ are different. The goal is to find a data subset $\mathcal{S} \subset \mathcal{T}$ of the training data pool $\mathcal{T}$ which maximizes the likelihood of the development data $\mathcal{D}$. A computationally feasible approach has to be developed.

In order to be able to investigate as many subsets of the data pool as possible, the calculation of the likelihood has to be fast. The naive and computationally expensive approach for likelihood calculation of an utterance given a HMM-based acoustic model would be to calculate the likelihood using the forward or backward algorithm whenever the data subset $\mathcal{S}$ for estimating new model parameters changes.

In the following it is shown that there is an alternative optimization criterion which can be calculated instantaneously only using the HMM sufficient statistics of the training and development data without explicit reconstruction of model parameters and re-calculation of the forced-alignment whenever the data subset changes. Furthermore, it is shown that an increase of the alternative optimization criterion necessarily implies an increase of the likelihood function.

Optimization Criterion The auxiliary Q-function (Eq. 2.20) is employed as optimization criterion for selective training. It is defined as the expectation of the log-likelihood function given initial model parameters. The same Q-function is also employed in the Expectation-Maximization (EM) framework [16] for iterative estimation of HMM parameters.

In the original definition of the Q-function the same data set $\mathcal{X}$ is employed for calculation of model parameters $\mathbf{\Theta}$ and calculation of the expected likelihood. Here, the following modification of the Q-function is considered:

$$Q(\mathbf{\Theta}, \hat{\mathbf{\Theta}}) = \sum_{\boldsymbol{z}} P(\boldsymbol{z}|\mathcal{D}, \mathbf{\Theta}) \log P(\boldsymbol{z}, \mathcal{D}|\hat{\mathbf{\Theta}}) \tag{4.1}$$

with symbols having the following meanings

- task-specific development data $\mathcal{D} = \{\boldsymbol{x}\}$
- data pool of training utterances $\mathcal{T} = \{\boldsymbol{u}^1, \boldsymbol{u}^2, \ldots\}$
- initial model parameters $\mathbf{\Theta}$ estimated on all training data $\mathcal{T}$
- parameters $\hat{\mathbf{\Theta}} = \{\hat{\mu}_{qm}, \hat{\sigma}_{qm}, \hat{w}_{qm}, \hat{a}_{qq'}\}$ estimated on the subset $\mathcal{S} \subset \mathcal{T}$
- state and mixture index sequence $\boldsymbol{z}$
- state index q
- mixture component index m
- mean $\hat{\mu}_{qm}$ of state q, mixture m
- variance $\hat{\sigma}_{qm}$ of state q, mixture m
- weight $\hat{w}_{qm}$ of state q, mixture m
- state occupation and transition probabilities $\hat{a}_{qq'}$

While the training data set $\mathcal{T}$ and its subset $\mathcal{S}$ are still employed for estimating initial $\mathbf{\Theta}$ and updated model parameters $\hat{\mathbf{\Theta}}$, the task-specific development data $\mathcal{D}$ are employed for calculating the expectation of the likelihood function.

Selection Condition The idea is to find a subset $\mathcal{S}$ of the training data pool $\mathcal{T}$ so that the model likelihood increases (Figure 4.2).

In the following it is shown that an increase of the Q-function, i.e. $Q(\hat{\mathbf{\Theta}}|\mathbf{\Theta}) > Q(\mathbf{\Theta}|\mathbf{\Theta})$, implies an increase of the likelihood, i.e. $P(\hat{\mathbf{\Theta}}) > P(\mathbf{\Theta})$. An increase of the Q-function is equivalent to the condition

$$\Delta Q = Q(\hat{\mathbf{\Theta}}|\mathbf{\Theta}) - Q(\mathbf{\Theta}|\mathbf{\Theta}) = \sum_{\boldsymbol{z}} P(\boldsymbol{z}|\mathcal{D}, \mathbf{\Theta}) \log \frac{P(\boldsymbol{z}, \mathcal{D}|\hat{\mathbf{\Theta}})}{P(\boldsymbol{z}, \mathcal{D}|\mathbf{\Theta})} > 0. \tag{4.2}$$

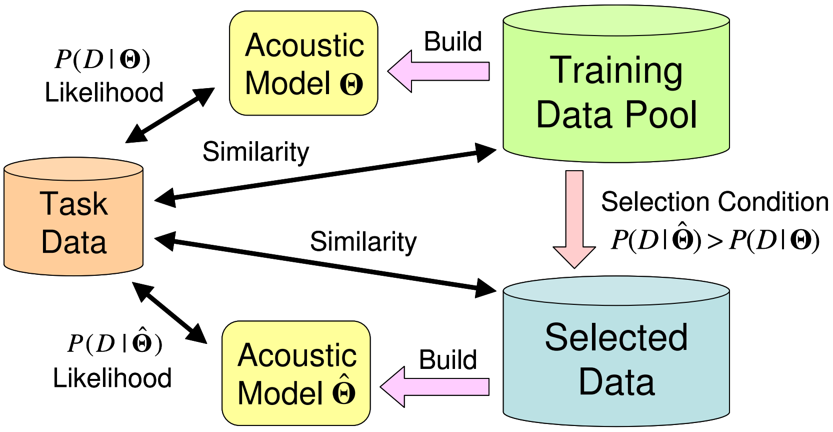
$P(D\,|\,\mathbf{\Theta})$
Likelihood
Acoustic Model $\mathbf{\Theta}$
Build
Training Data Pool
Similarity
Task Data
Selection Condition
$P(D\,|\,\hat{\mathbf{\Theta}}) > P(D\,|\,\mathbf{\Theta})$
Similarity
$P(D\,|\,\hat{\mathbf{\Theta}})$
Likelihood
Acoustic Model $\hat{\mathbf{\Theta}}$
Build
Selected Data

Figure 4.2. Likelihood-based selection condition.

This expression can be transformed to

$$\Delta Q = -D(P(\boldsymbol{z}|\mathcal{D},\mathbf{\Theta}) \parallel P(\boldsymbol{z}|\mathcal{D},\hat{\mathbf{\Theta}})) + \log \frac{P(\mathcal{D}|\hat{\mathbf{\Theta}})}{P(\mathcal{D}|\mathbf{\Theta})} \sum_{\boldsymbol{z}} P(\boldsymbol{z}|\mathcal{D},\mathbf{\Theta}) > 0. \qquad (4.3)$$

Since the KL distance is always positive, the first term is negative and since the sum over the probabilities $P(\boldsymbol{z}|\mathcal{D},\mathbf{\Theta})$ is positive it follows that

$$P(\mathcal{D}|\hat{\mathbf{\Theta}}) > P(\mathcal{D}|\mathbf{\Theta}), \qquad (4.4)$$

i.e. the likelihood increases. This proves the initial statement.

Instantaneous Calculation Instantaneous calculation of the Q-function is possible using sufficient statistics (Section 2.2.2). Calculation of the likelihood is only required once for the initial model parameters.

For simplicity of notation, $\boldsymbol{x} = (\ldots, x_t, \ldots)$ and $\boldsymbol{u} = (\ldots, u_t, \ldots)$ are assumed to be one-dimensional. Nevertheless, it is easy to define equations for multivariate data. In Section 2.3, Eq. 2.24 it was shown that the auxiliary Q-function for a HMM with GMM parameters $\hat{\mu}_{qm}, \hat{\sigma}_{qm}$ is proportional to

$$\begin{aligned} Q(\hat{\mathbf{\Theta}}|\mathbf{\Theta}) \quad &\propto \quad \sum_{q'}\sum_{q}\sum_{t} \gamma_{q'\to q}(t) \log \hat{a}_{q'q} \\ &+ \quad \sum_{q}\sum_{m}\sum_{t} \gamma_{qm}(t) \left[\log \hat{w}_{qm} - \frac{1}{2}\log \hat{\sigma}_{qm} - \frac{1}{2}(x_t - \hat{\mu}_{qm})^2 \frac{1}{\hat{\sigma}_{qm}} \right]. \end{aligned}$$

Reorganizing terms by product expansion

$$\begin{aligned}
Q(\hat{\mathbf{\Theta}}|\mathbf{\Theta}) &\propto \sum_{q'}\sum_{q}\sum_{t}\gamma_{q'\to q}(t)\log\hat{a}_{q'q} \\
&+ \sum_{q}\sum_{m}\sum_{t}\gamma_{qm}(t)\left[\log\hat{w}_{qm}-\frac{1}{2}\log\hat{\sigma}_{qm}\right] \\
&- \sum_{q}\sum_{m}\sum_{t}\gamma_{qm}(t)\left[\frac{x_t^2-2x_t\hat{\mu}_{qm}+\hat{\mu}_{qm}^2}{2\hat{\sigma}_{qm}}\right]
\end{aligned}$$

and changing the location of the summation over time index t the expression can be transformed to

$$\begin{aligned}
&\propto \sum_{q'}\sum_{q}\left\{\log\hat{a}_{q'q}\sum_{t}\gamma_{q'\to q}(t)\right\} \\
&+ \sum_{q}\sum_{m}\left\{\sum_{t}\gamma_{qm}(t)\log\hat{w}_{qm}-\sum_{t}\gamma_{qm}(t)\frac{\log\hat{\sigma}_{qm}}{2}\right\} \\
&- \sum_{q}\sum_{m}\left\{\frac{\sum_{t}\gamma_{qm}(t)x_t^2-2\hat{\mu}_{qm}\sum_{t}\gamma_{qm}(t)x_t+\hat{\mu}_{qm}^2\sum_{t}\gamma_{qm}(t)}{2\hat{\sigma}_{qm}}\right\}.
\end{aligned}$$

It is obvious that the summations over time index t can be precalculated and that they are independent from the parameters $\hat{\mathbf{\Theta}}$ estimated on the training data. If we define the sufficient statistics (SS) for the development data $\mathcal{D}$ as

$$\begin{aligned}
v_{q'q} &= \sum_{t}\gamma_{q'\to q}(t) \\
f_{qm} &= \sum_{t}\gamma_{qm}(t) \\
y_{qm} &= \sum_{t}\gamma_{qm}(t)\ x_t \\
z_{qm} &= \sum_{t}\gamma_{qm}(t)\ x_t^2
\end{aligned}$$

the Q-function can be simplified to

$$\begin{aligned}
&\propto \sum_{q'}\sum_{q}\{v_{q'q}\log\hat{a}_{q'q}\} \\
&+ \sum_{q}\sum_{m}\left\{f_{qm}\log\hat{w}_{qm}-\frac{f_{qm}}{2}\log\hat{\sigma}_{qm}\right\} \\
&- \sum_{q}\sum_{m}\left\{\frac{z_{qm}-2\hat{\mu}_{qm}y_{qm}+\hat{\mu}_{qm}^2f_{qm}}{2\hat{\sigma}_{qm}}\right\}.
\end{aligned}$$

In the following derivation the transition probability term is omitted for the sake of simplicity. The SS for any subset of the training data $\mathcal{S} \subset \mathcal{T}$ are given by the equations

$$\begin{aligned}
c_{qm} &= \sum_i c^i_{qm} &&= \sum_i \sum_t \gamma^i_{qm}(t) \\
\nu_{qm} &= \sum_i \nu^i_{qm} &&= \sum_i \sum_t \gamma^i_{qm}(t)\ u^i_t \\
\xi_{qm} &= \sum_i \xi^i_{qm} &&= \sum_i \sum_t \gamma^i_{qm}(t)\ (u^i_t)^2
\end{aligned}$$

An estimate for the model parameters $\hat{\mathbf{\Theta}}$ can be obtained directly from the sum of utterance-based SS $\mathbf{S}_i = (c^i_{qm}, \nu^i_{qm}, \xi^i_{qm})^T$. The formulas for the reconstruction of new model parameters

$$\hat{\mathbf{\Theta}} = h(\sum_i \mathbf{S}_i) \tag{4.5}$$

for any subset of training utterances $\mathcal{S} = \{\mathbf{u}_i\}$ are

$$\begin{aligned}
\hat{w}_{qm} &= \frac{\sum_i c^i_{qm}}{\sum_n \sum_i c^i_{qn}} &&= \frac{c_{qm}}{\sum_n c_{qn}} \\
\hat{\mu}_{qm} &= \frac{\sum_i \nu^i_{qm}}{\sum_i c^i_{qm}} &&= \frac{\nu_{qm}}{c_{qm}} \\
\hat{\sigma}_{qm} &= \frac{\sum_i \xi^i_{qm}}{\sum_i c^i_{qm}} - \hat{\mu}^2_{qm} &&= \frac{\xi_{qm}}{c_{qm}} - \frac{\nu^2_{qm}}{c^2_{qm}}
\end{aligned}$$

Since the SS are decomposable w.r.t. the training utterances $\mathbf{u}_i$, removing utterances from or adding utterances to the subset $\mathcal{S}$ means subtracting or adding the corresponding SS $\boldsymbol{S}_i$. When substituting the SS of the training data subset for the estimated parameters $\hat{\mathbf{\Theta}}$ and reorganizing terms we obtain the equation

$$\begin{aligned}
\propto \sum_q \sum_m \left\{ f_{qm} \log \left[\frac{c_{qm}}{\sum_n c_{qn}} \right] - \frac{f_{qm}}{2} \log \left[\frac{\xi_{qm} c_{qm} - \nu^2_{qm}}{c^2_{qm}} \right] \right\} \\
- \sum_q \sum_m \left\{ \frac{z_{qm} c^2_{qm} - 2\nu_{qm} y_{qm} c_{qm} + \nu^2_{qm} f_{qm}}{2\xi_{qm} c_{qm} - 2\nu^2_{qm}} \right\}
\end{aligned}$$

This shows that the Q-function can be calculated directly from the SS of the development and the SS of the training data. The computational complexity depends on the number of model parameters, i.e. number of states and mixture components, but it is independent from the size of the development data.

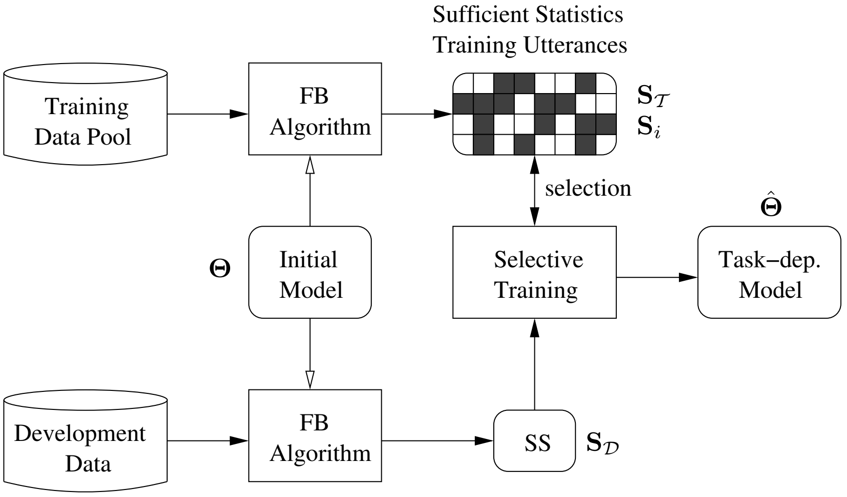

Figure 4.3. Preparations and setup required for selective training. After calculating the sufficient statistics (SS) for each training utterance $\boldsymbol{S}_i$, the SS for the whole data pool $\boldsymbol{S}_{\mathcal{T}}$ and the SS for the whole development data $\boldsymbol{S}_{\mathcal{D}}$, the selective training algorithm *ST_DelScan* or *ST_DelAdd* is applied.

4.2.3 Selection Algorithm

In the previous section it has been shown that the calculation of the Q-function only depends on the sufficient statistics (SS) of the training and the development data w.r.t. the initial model parameters $\mathbf{\Theta}$. The feasibility of selective training is now obvious. Selective training works by successively adding or subtracting the SS of single training utterances. This means modifying $\hat{\mathbf{\Theta}}$ so that the Q-function's value increases.

Since there are too many possibilities (2^n) to select a subset from a data pool with n utterances, only a very small number of possible data subsets can be investigated in practice. It is important to employ a selection strategy which is able to identify an appropriate data subset with a limited number of tests.

Figure 4.3 depicts the overall setup and preparations before selective training (ST) is carried out. There are several possibilities to define a concrete ST algorithm. Here, two variants are considered: The delete scan algorithm *ST_DelScan*, which tests every training utterance only once for deletion, and the *ST_DelAdd* algorithm which successively deletes and adds utterances for several iterations.

The *ST_DelScan* variant works as follows:

1. Let R be the set of all (selected) training utterances.

2. Obtain $\{\mathbf{S}_i\}$, the SS of each training utterance $\mathbf{u}_i$.

3. Obtain $\mathbf{S}_{\mathcal{D}}$, the SS of the whole development data.

4. Obtain $\mathbf{S}_{\mathcal{T}}$, the SS of the whole training data.

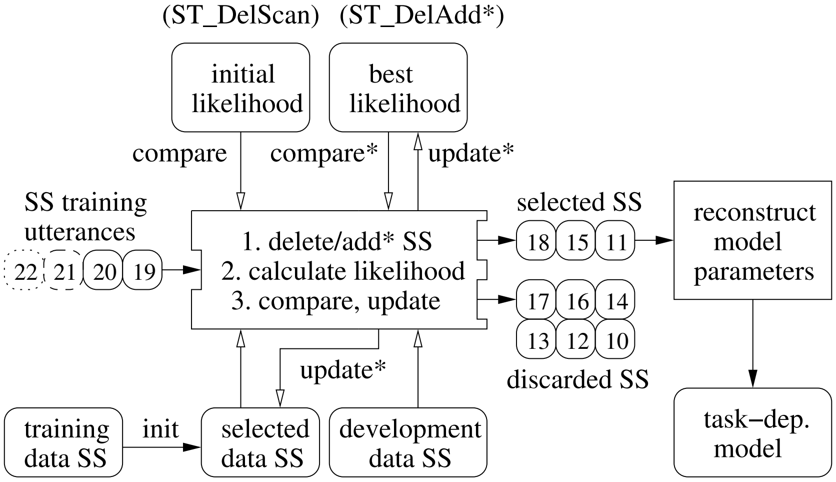

Figure 4.4. Detailed illustration of the selective training procedure. Actions marked with (*) are only carried out by the *ST_DelAdd* variant. The sufficient statistics (SS) of each training utterance are processed one by one in series. An utterance is only selected for parameter estimation if the model likelihood increases. While *ST_DelScan* considers each utterance only once for deletion, *ST_DelAdd* examines each discarded utterance again for addition.

5. Evaluate $q := Q(h(\mathbf{S}_{\mathcal{T}}), \mathbf{\Theta})$.

6. For each utterance $\mathbf{u}_i \in R$ do:

   a. Evaluate $q' := Q(h(\mathbf{S}_{\mathcal{T}} - \mathbf{S}_i), \mathbf{\Theta})$.

   b. If $q' > q$, then discard utterance $\mathbf{u}_i$: $R := R - \{\mathbf{u}_i\}$

7. Use $\hat{\mathbf{\Theta}} = h(\sum\limits_{\mathbf{u}_i \in R} \mathbf{S}_i)$ as new model parameters.

8. Retrain with utterance set R for several iterations.

The idea is, that if the independent deletion of single training utterances leads to an increase of model likelihood, it should not be used for training. Consequently, the decision to discard one utterance is independent from the deletion of a previous or following utterance.

Instead of (8.) simple retraining for several iterations, it would be better to determine the utterance subset separately for every training iterations. The graphical illustration of the algorithm in Section 4.2.4 will show that the set of selected data changes significantly if there is a large mismatch between the development data and the initial model. Retraining with the same subset in case of model to data mismatch would distract selective training from a reasonable solution. However, data selection at each iteration would be far more computationally intensive.

Instead of considering every utterance only once for deletion, step (6.) could also be carried out iteratively, while alternating between deleting (already) selected utterances or adding unselected utterances. This is realized in the *ST_DelAdd* variant of the algorithm. Step (6.) has to be modified as follows:

6. Repeat for a predefined number of iterations:
   - I. For each $\mathbf{u}_i \in R$ do:
     - a. Evaluate $q' := Q(h(\mathbf{S}_{\mathcal{T}} - \mathbf{S}_i), \mathbf{\Theta})$.
     - b. If $q' > q$, then discard utterance $\mathbf{u}_i$:
$R := R - \{\mathbf{u}_i\}$, $\mathbf{S}_{\mathcal{T}} := \mathbf{S}_{\mathcal{T}} - \mathbf{S}_i$ and $q := q'$
   - II. For each $\mathbf{u}_i \notin R$ do:
     - a. Evaluate $q' := Q(h(\mathbf{S}_{\mathcal{T}} + \mathbf{S}_i), \mathbf{\Theta})$.
     - b. If $q' > q$, then remember utterance $\mathbf{u}_i$:
$R := R \cup \{\mathbf{u}_i\}$, $\mathbf{S}_{\mathcal{T}} := \mathbf{S}_{\mathcal{T}} + \mathbf{S}_i$ and $q := q'$

A drawback of this approach is, that the decision to delete or add an utterances depends on the order of presenting training utterances to the algorithm. Furthermore, On the other hand, the value of the auxiliary Q-function can increase more than in case of the *ST_DelScan* variant. Figure 4.4 shows the processing steps of both variants of the selective training algorithm.

Convergence The question whether a method for parameter estimation converges is always of importance. The selection condition and algorithm are designed so that the $Q(\hat{\mathbf{\Theta}}|\mathbf{\Theta})$-function and the likelihood $P(\mathcal{D}|\hat{\mathbf{\Theta}})$ increase. Both functions reach their maximum by definition at the ML solution of the parameters $\mathbf{\Theta}$ estimated on the development data $\mathcal{D}$ set.

$$\hat{\mathbf{\Theta}}_{\mathcal{D}}^{ML} = \arg\max_{\hat{\mathbf{\Theta}}} P(\mathcal{D}|\hat{\mathbf{\Theta}}) \tag{4.6}$$

$$\hat{\mathbf{\Theta}}_{\mathcal{D}}^{ML} = \arg\max_{\hat{\mathbf{\Theta}}} Q(\hat{\mathbf{\Theta}}|\hat{\mathbf{\Theta}}) = \arg\max_{\hat{\mathbf{\Theta}}} \sum_{\boldsymbol{z}} P(\boldsymbol{z}|\mathcal{D},\hat{\mathbf{\Theta}}) \log P(\boldsymbol{z},\mathcal{D}|\hat{\mathbf{\Theta}}) \tag{4.7}$$

The maximum could in principle be reached by repeated iterations of the selective training procedure, because $Q(\mathbf{\Theta}^{[j]}|\mathbf{\Theta}^{[j-1]}) > Q(\mathbf{\Theta}^{[j-1]}|\mathbf{\Theta}^{[j-1]})$. However, there is the problem of local minima and there is no guarantee that $Q(\mathbf{\Theta}^{[j]}|\mathbf{\Theta}^{[j]}) \geq Q(\mathbf{\Theta}^{[j-1]}|\mathbf{\Theta}^{[j-1]})$, because $Q(\mathbf{\Theta}^{[j]}|\mathbf{\Theta}^{[j]})$ may be smaller than $Q(\mathbf{\Theta}^{[j]}|\mathbf{\Theta}^{[j-1]})$ in general.

4.2.4 Graphical Illustration of Selective Training

In this section the effectivity of selective training is shown by simulating its behavior with artificially generated data. For simplicity a mixture model with Gaussian components and two-dimensional feature vectors is employed. The task-specific

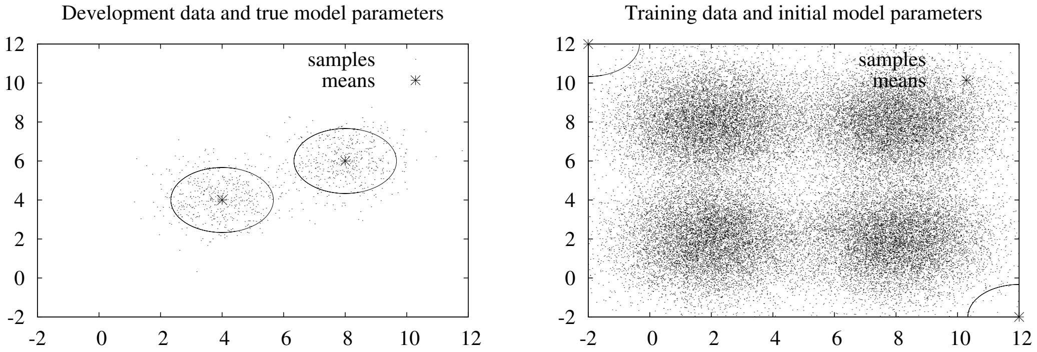

Figure 4.5. Left: Development data $\mathcal{D}$ and the GMM employed for its random generation. Right: Training data pool and initial model parameters $\mathcal{T}$.

development data set contains 500 samples each from the components of a mixture model with the following parameters:

$$\boldsymbol{w} = (0.5, 0.5), \quad \boldsymbol{\mu}_1 = (4.0, 4.0), \quad \boldsymbol{\mu}_2 = (8.0, 6.0), \quad \boldsymbol{\Sigma}_1 = \boldsymbol{\Sigma}_2 = \begin{bmatrix} 1.0 & 0.0 \\ 0.0 & 1.0 \end{bmatrix}.$$

The data pool has been generated by a mixture model with the following four components:

$$\boldsymbol{\mu}_1 = (2.0, 2.0), \quad \boldsymbol{\mu}_2 = (2.0, 8.0), \quad \boldsymbol{\mu}_3 = (8.0, 2.0), \quad \boldsymbol{\mu}_4 = (8.0, 8.0),$$
$$\boldsymbol{w} = (0.25, 0.25, 0.25, 0.25), \quad \boldsymbol{\Sigma}_1 = \boldsymbol{\Sigma}_2 = \boldsymbol{\Sigma}_3 = \boldsymbol{\Sigma}_4 = \begin{bmatrix} 1.5 & 0.0 \\ 0.0 & 1.5 \end{bmatrix}.$$

There are 10,000 sample points per mixture component. Sample points have been grouped so that there are ten sample points per element of the training data set. A plot of both data sets, the ideal model and the initial model are shown in Figure 4.5. The initial model parameters were selected so that there is no severe overlap with the development data or training data pool. This makes it possible to show that selection works even in case of a bad initialization which is usually very important for algorithms of the EM framework.

Both variants of the selective training procedure are simulated using these data and this initial model. In case of the *ST_DelScan* algorithm, three delete scan steps are employed in each selection iterations. For the *ST_DelAdd* algorithm, five interleaved, successive deletion/addition steps are employed in each selection iteration. Selected data and updated model parameters after the 1st, 5th, 10th and 20th retraining iteration are shown in Figure 4.6. The distribution of the development and the selected data appear to be quite different for the first iterations, especially when employing the delete scan algorithm.

Comparing model parameters as selective training progresses it is obvious that *ST_DelAdd* has converged after about 10 training iterations while *ST_DelScan* required up to 20 training iterations. As the number of iterations increase, there seems to be a higher resemblance of the selected data to the development data.

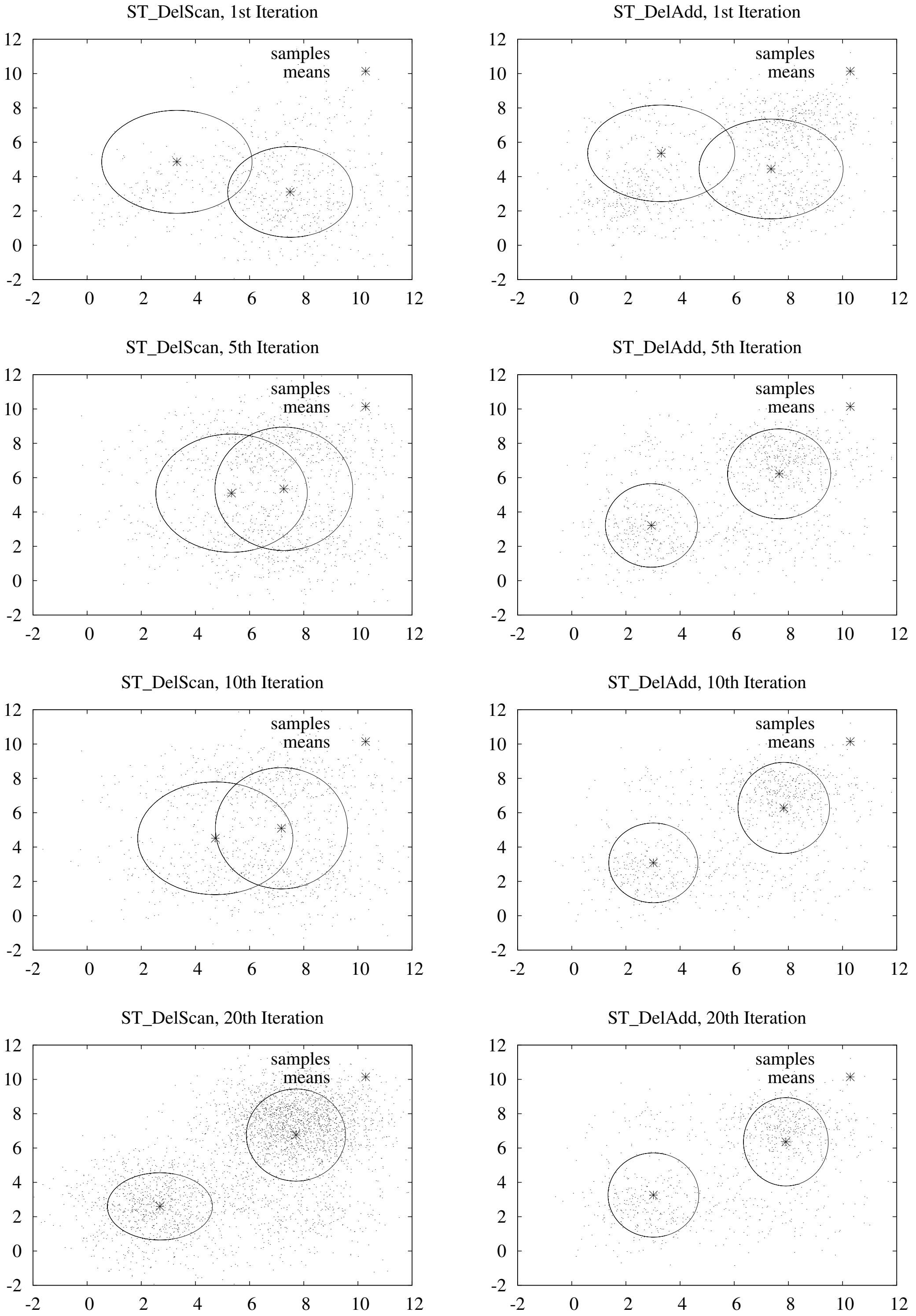

Figure 4.6. From top to bottom: Selected data and updated model parameters after the 1st, 5th, 10th and 20th selective training iteration (left: *ST_DelScan*, right: *ST_DelAdd*).

Table 4.1. Updated model parameters after each selective training iteration

Algo	*ST_DelScan*				*ST_DelAdd*			
Iter	$\hat{\boldsymbol{\mu}}_1$	$\hat{\boldsymbol{\mu}}_2$	diag $\hat{\boldsymbol{\Sigma}}_1$	diag $\hat{\boldsymbol{\Sigma}}_2$	$\hat{\boldsymbol{\mu}}_1$	$\hat{\boldsymbol{\mu}}_2$	diag $\hat{\boldsymbol{\Sigma}}_1$	diag $\hat{\boldsymbol{\Sigma}}_2$
0	(-2,12)	(12,-2)	(2.0,2.0)	(2.0,2.0)	(-2,12)	(12,-2)	(2.0,2.0)	(2.0,2.0)
1	(3.3,5.4)	(7.4,4.4)	(5.3,5.7)	(5.1,6.1)	(3.3,4.7)	(7.5,3.1)	(5.5,6.5)	(3.0,5.0)
2	(3.3,3.9)	(7.3,5.8)	(5.0,5.6)	(4.4,5.9)	(5.8,7.3)	(7.6,5.0)	(2.7,5.6)	(2.9,8.7)
3	(3.0,3.4)	(7.5,6.0)	(3.6,4.8)	(3.6,5.5)	(5.2,6.8)	(7.4,6.8)	(5.2,6.9)	(6.1,6.5)
5	(2.9,3.2)	(7.7,6.2)	(2.1,4.3)	(2.6,4.9)	(5.3,5.1)	(7.3,5.3)	(5.6,8.6)	(4.7,9.3)
10	(3.0,3.1)	(7.8,6.3)	(2.0,3.9)	(2.1,5.1)	(4.7,4.5)	(7.2,5.1)	(5.9,7.8)	(4.3,9.0)
20	(3.0,3.3)	(7.9,6.4)	(2.0,4.3)	(1.8,4.8)	(2.7,2.6)	(7.7,6.8)	(2.7,2.8)	(2.4,5.2)
true	(4.0,4.0)	(8.0,6.0)	(1.0,1.0)	(1.0,1.0)	(4.0,4.0)	(8.0,6.0)	(1.0,1.0)	(1.0,1.0)

From the updated parameters after each iteration (Table 4.1), it is clear that the selective training algorithm converges to a different solution than the true model, which has employed to generate the development data. The final solution is a trade-off between the distribution of the development data and a certain subset of the training data pool. The scatter of the selected data is also larger than that of the development data. The solution when using *ST_DelScan* is more influenced by the distribution of the training data than when using *ST_DelAdd*.

This behavior of the selective algorithm is actually desired. If selective training would converge to the same solution (same mean vectors, same scatter) as EM training using only the development data it would be useless. It is clear that selective training can extract a subset from the data pool which similar to the development data. Consequently, the estimated means are close to that of the development data and a higher scatter enables the model to also consider the training data distribution in a larger region around the development data.

The Q-function's value immediately before and after each selective training iteration is shown in Table 4.2. Although there is no obvious tendency within an iteration in case of the *ST_DelScan* algorithm, the Q-function's value seems to increase after several iterations. *ST_DelAdd* is more well-behaved, since the Q-function's value obviously increases after each iteration. This result is in concordance with the already observed faster convergence of the *ST_DelAdd* algorithm.

Finally, the outcome of EM training using only the development data and selective training using the development data and the large data pool should be compared. The left graph of Figure 4.7 shows the development data and the model parameters after the 5th training iteration, the right graph the selected data and model parameters after the 5th selective training iteration when using the *ST_DelAdd* algorithm. While the location of the means is quite similar, the variance is larger when using selective training. This shows that the model estimated using selective training has the tendency to be more general than the model estimated only on the development data. To combine development and selected data for final retraining is recommendable.

Table 4.2. Q-function's value before and after each selective training iteration

Algorithm	*ST_DelScan*		*ST_DelAdd*	
Iteration	$Q(\boldsymbol{\Theta}\|\boldsymbol{\Theta})$	$Q(\hat{\boldsymbol{\Theta}}\|\boldsymbol{\Theta})$	$Q(\boldsymbol{\Theta}\|\boldsymbol{\Theta})$	$Q(\hat{\boldsymbol{\Theta}}\|\boldsymbol{\Theta})$
1st	-22,965	-3,178	-22,965	-2,942
2nd	-3,292	-3,519	-3,051	-2,976
3rd	-3,552	-3,318	-2,863	-2,844
5th	-3,198	-3,257	-2,607	-2,579
10th	-3,165	-3,168	-2,489	-2,489
20th	-2,725	-2,723	-2,446	-2,447

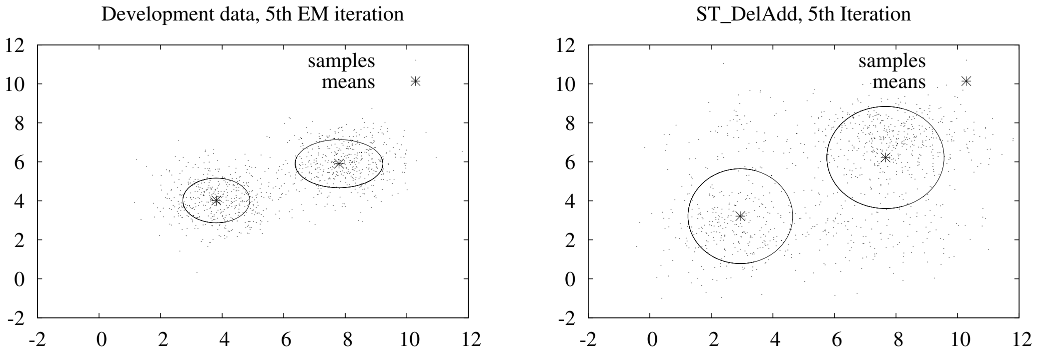

Figure 4.7. Left: 5th EM iteration with development data. Right: 5th selective training iteration (*ST_DelAdd*).

4.2.5 Remarks and Conclusion

From the graphical simulation of selective training with artificial data it is clear that the convergence of *ST_DelAdd* is faster than *ST_DelScan*. Although the selection criterion is based on the maximum likelihood principle, there were no signs of overtraining. This is due to the circumstance that training and development data set are disjoint.

It could also be observed that the set of selected data changes after each training iteration. Consequently, retraining with the same set of selected data for several iterations is invalid. Ignoring this would distract selective training from the true solution. However, there were almost no changes after model parameters converged to a final solution.

Since the *ST_DelScan* algorithm is parallelizable, it will be employed for most selective training experiments in Chapter 5 to reduce computation time. To further save computational costs, the initially set of selected data is employed for several retraining iterations. In order to minimize the problem of convergence, initial model parameters should be estimated on the whole data pool, or the initial model should be adapted with the development data set.

4.3 Possible Applications

There are several possible applications for the proposed selective training framework. Three are mentioned in the following. One application, the original idea of selective training, is the cost-effective construction of task-adapted acoustic models using existing speech data resources. Given a large pool of inhomogeneous speech data it is possible to extract a certain data subset, e.g. only children speech or only adult speech. For extraction only a relatively small amount of task-specific example data would be required. The extracted data can then be employed to construct a task-adapted acoustic model (Figure 4.8).

A further application is the combination of selective and unsupervised training. The initial data pool may contain noisy speech data of low quality, noise-only data, non-verbals such as coughing or laughing, etc. It would be easy to exclude these data if they are human-labeled. However, human transcriptions are expensive. A cost reduction would be achieved if the acoustic model could be constructed from automatically collected speech data in unsupervised manner. However, unsupervised model training requires some kind of label for the training utterances. In order to remove noisy data or automatically transcribed utterances with a wrong transcription, selective training can be employed. The development data set would consist of human-labeled speech data of good quality (Figure 4.9).

Moreover, selective training could be employed for cross-language acoustic modeling. If there is no speech data for a certain language available, it is worth considering to combine the speech databases of other languages in order to cover the phoneme set of the desired target language. Phonemes of several source languages with speech corpora available could be mapped to the phonemes of a target language. Automatic selection of appropriate training data could be achieved by maximizing the likelihood of a small set of development data for the target language (Figure 4.10).

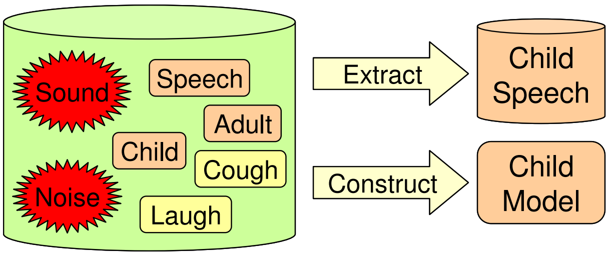

Figure 4.8. Selective training could be applied to extract a certain subset of task-specific speech data from a large data pool in order to construct a task-adapted acoustic model.

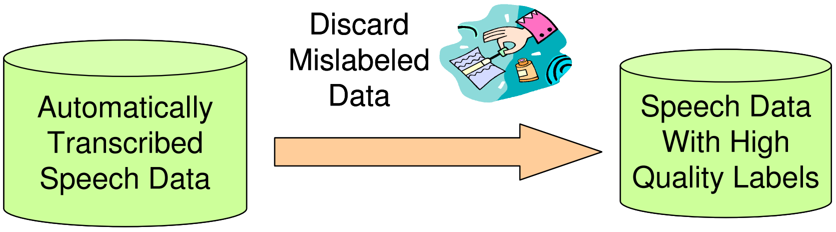
Automatically Transcribed Speech Data
Discard Mislabeled Data
Speech Data With High Quality Labels

Figure 4.9. Selective training could be employed to discard utterances with wrong automatic transcriptions to improve unsupervised model training.

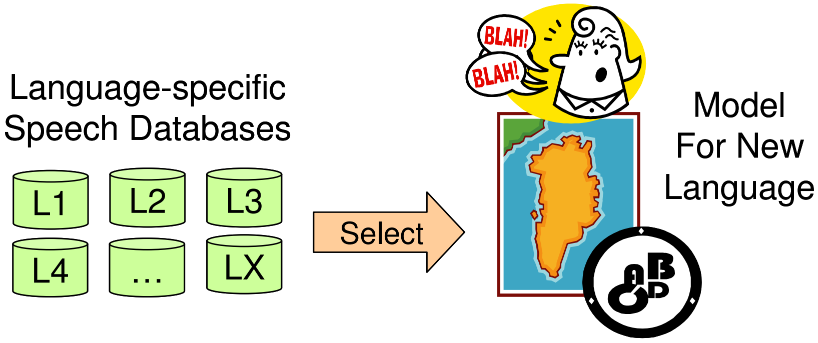

Figure 4.10. Selective training could also be applied to construct acoustic models for a new language with almost no speech data available by selecting appropriate data from existing resources of other languages.

Chapter 5

Experiments with Selective Training

5.1 Human-Transcribed Data Pool

Selective training is applied to construct an acoustic model for two speaker groups, preschool children and elderly persons, for which data collection is more difficult in general. For example, among the data collected with the real-environment guidance system *Takemaru* during the first two years, only 10% of the utterances are from preschool children and less than one thousand utterances ($\leq$ 1%) are from elderly people. Furthermore, the word accuracy for spontaneous preschool children speech with standard adult or child acoustic models for dictation is only about 10-20%. Therefore, the augmentation of preschool training data with school children utterances and elderly training data with adult utterances is worth considering to improve ASR performance.

Table 5.1 gives details about the speech data employed for training and evaluation in the following experiments. The data has been collected with the speech-oriented guidance system Takemaru-kun [53] (cf. Chapter 3.3).

None of the test utterances is part of the training or task-specific development data employed for selective training of the acoustic model. The perplexity of the test set (5,742 words) in experiment A is 8.3 for a language model trained

Table 5.1. Training data pool and task-specific development data. The purpose of experiment (A) is to obtain a preschool-adapted AM by selecting utterances from an elementary school children speech data pool. In experiment (B), utterances from adult speakers are selected to build a better AM for elderly speech

	Training Data Pool		Development Set		Test Set
Exp	Group	#utr / time	Group	#utr / time	#utr / time
(A)	Element.	29,776 / 17 h	Preschool	500 / 17 m	1.5k / 53 m
(B)	Adult	17,874 / 9 h	Elderly	53 / 2 m	400 / 12 m

Table 5.2. Conditions for evaluating the two variants of selective training

AM Training	HTK 3.2 [31]
LM Training	Palmkit 1.0.31 [58]
Acoustic Features	12 MFCC, 12 Δ MFCC, Δ E
Language Model	3-gram, Witten-Bell Smoothing
ASR Engine	Julius 3.4 [37]

on transcriptions of preschool utterances including the transcriptions of the test data. For experiment B, the language model is trained on transcriptions of adult utterances, i.e. it is open w.r.t. the test data. The perplexity of the corresponding test set (1,609 words) is 16.3.

5.1.1 Comparison of Algorithm Variants

In the following, both variants of the ST algorithm as described in Section 4.2.3 are evaluated. The initial acoustic model is obtained from scratch by training with all utterances in the data pool. There is one 3-state HMM each for 35 phonemes and three silence models. Each HMM state has up to 16 Gaussian mixture densities. Covariance matrices are diagonal. The sufficient statistics are calculated with this initial all data model. Only one delete scan iteration is conducted for the *ST_DelScan* algorithm. Deletion and addition of utterances is repeated five times for the *ST_DelAdd* algorithm. In order to prevent flooring of variances, a threshold of 200 is set for the minimum number of examples required per phoneme. Other experimental conditions are given in Table 5.2.

Experimental Results and Discussion Tables 5.3 and 5.4 show the result for building a preschool (A) and elderly-dependent (B) acoustic model, respectively. Both variants of the ST algorithm are compared to model training without data selection, i.e. employ all utterances in the data pool for retraining. In case of the preschool experiment (A), there is only an improvement of 1.3% absolute (2.8% relative) over the initial model when retraining the initial model with all utterances in the data pool. However, with selection the performance increases up to 5.1% absolute (11.0% relative). Although the improvement gain is highest after the first iteration, retraining with the selected set of utterances for several times leads to further improvements in word accuracy.

The same can be observed in experiment (B) for building an elderly-dependent model using adult speech. An increase of up to 2.8% absolute (3.9% relative) in recognition accuracy by selective training versus almost no improvement without selection. The difference in performance between the two variants of the ST algorithm in both experiments is rather small. Statistics about the number of utterances selected are given in Table 5.5. While the *ST_DelScan* variant selects more than one third of the utterances in the data pool, only about 10-20% are extracted by the *ST_DelAdd* variant.

Table 5.3. Comparison of model retraining with data selection versus retraining without data selection (Preschool Speech, Word Accuracy in %)

Experiment A	Training Iteration					
Algorithm	init	1	2	3	4	5
No Selection	46.4	46.7	47.3	**47.4**	47.3	47.3
ST_DelScan	46.4	50.4	50.6	50.8	**51.4**	51.3
ST_DelAdd	46.4	50.5	51.1	51.2	51.2	**51.5**

Table 5.4. Comparison of model retraining with data selection versus retraining without data selection (Elderly Speech, Word Accuracy in %)

Experiment B	Training Iteration					
Algorithm	init	1	2	3	4	5
No Selection	72.3	72.2	72.0	72.4	**72.5**	72.1
ST_DelScan	72.3	74.4	**75.1**	74.7	74.6	74.5
ST_DelAdd	72.3	73.5	73.8	73.8	**74.1**	73.7

From Figure 5.1 it is clear that rather utterances with a lower model likelihood are selected. Nevertheless, there is much overlap between the likelihood distributions of selected and discarded utterances, so that a simple selection rule such as "select all utterances with a likelihood below a threshold" would be far less effective than the proposed ST algorithm. The same tendency has been observed in the experiments (A) and (B).

How the value of the Q-function changes is depicted in Figure 5.2. The largest increase can be observed during deleting utterances in the first and second iteration. The number of discarded (-) and added (+) utterances during the five iterations was: -11,715, +203; -2,870, +44; -341, +21; -42, +1; -10, +0. Almost nothing is gained when adding previously deleted utterances again. Consequently, the addition step of *ST_DelAdd* could also be omitted.

Table 5.6 shows the performance of selective training depending on the development set size. There is already an improvement with only five development utterances. Maximum possible performance seems to be reached with about 100 utterances. The number of selected utterances does not depend much on the amount of task-specific development data.

Finally, selective training is compared to adaptation with the task-specific

Table 5.5. Number and percentage of utterances chosen from the data pool by the proposed selective training algorithm

Experiment A		Experiment B	
ST_DelScan	*ST_DelAdd*	*ST_DelScan*	*ST_DelAdd*
10,697 (36%)	4,299 (14%)	7,704 (43%)	3,165 (18%)

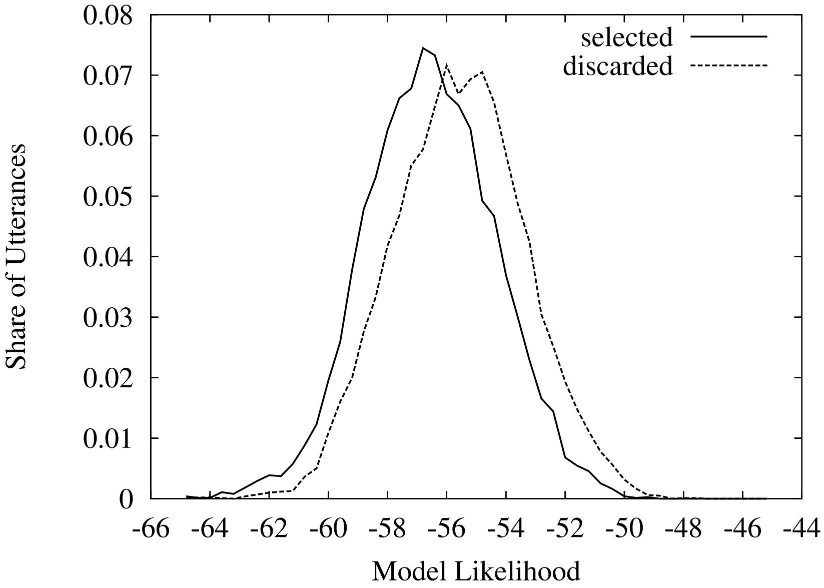

Figure 5.1. Likelihood distribution of the initial model given the selected and the discarded training utterances (Experiment B, *ST_DelScan*).

Table 5.6. Relationship between the number of utterances in the development set and the performance (word accuracy in %) of selective training. (Experiment A, *ST_DelScan*)

Development Set Size	5	10	20	50	100
ST (1st iteration)	49.1	49.3	50.4	50.4	50.5
ST (5th iteration)	49.6	50.3	50.2	50.8	51.2
# selected utterances	9,633	8,715	9,393	9,617	10,287

development data. A standard adaptation method for relatively few adaptation data is Maximum Likelihood Linear Regression (MLLR) [48]. The performance of MLLR is obtained by considering the best result among evaluation for 2, 4, 8, 16 and 32 regression classes and two adaptation iterations. All utterances in the development set (500 utterances for experiment A, 53 utterances for experiment B) are used for adaptation. The result in Figure 5.3 shows, that selective training is superior to both MLLR adaptation and retraining without selection. The advantage of selective training over MLLR also is that it does not require to set a parameter such as the number of regression classes.

Speaker-adaptive training (SAT) [3] is an acoustic model training framework to couple speaker-dependency from phonetic variation. By transforming each speaker's training data separately, the resulting model is normalized with respect to speaker characteristics. It is reported, that a model obtained by the SAT framework is a better seed model for adaptation than a speaker-independent model built by conventional Baum-Welch training. However, this is impossible in

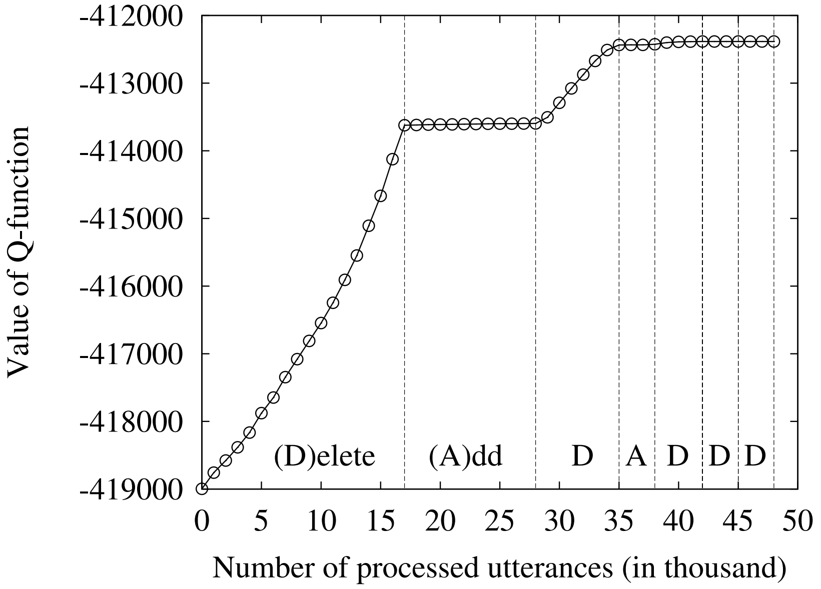

Figure 5.2. Behavior of the Q-function (Experiment B, *ST_DelAdd*).

case of the Takemaru database, which consists of utterances from a large number of unknown speakers, and with no speaker label available. Here it has also to be mentioned, that the proposed method for selective training optimizes all HMM parameters with respect to the task-specific data. Consequently, the risk that the final task-dependent model has flat and overlapping densities because of large variances is small.

Computational Requirements This section gives information about the computational requirements in (disk) space and (CPU) time for selective training. The time to extract the sufficient statistics (SS) for each training utterance is the same as for conventional Baum-Welch training. Additional time is only needed for storing the SS and running the ST algorithm. Reconstruction of model parameters is possible within milliseconds. Rather than CPU time, physical disk space and data transfer rate are important issues. The size of the SS is proportional to the number of model parameters. Consequently, much more disk space is needed to store the SS in comparison to the feature vector sequence or the discrete time speech signal. Fortunately, an utterance usually contains only a small subset of all target language phonemes. This means that most SS are zero. Hence, a high compression ratio (e.g. 1:5 for experiment A) can be achieved for most utterances.

Table 5.7 shows the run time and disk space required for conducting experiments A and B. A state-of-the-art personal computer with a 3.2 GHz CPU was employed. The selective training procedure took only about 20 minutes for experiment A, and 27 minutes for experiment B. Most of the CPU time is used to evaluate the optimization criterion (Q-function). The disk space required to store

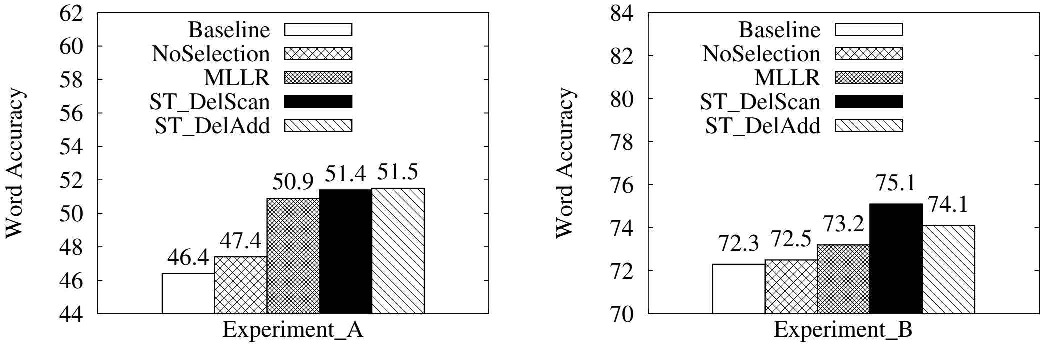

Figure 5.3. Comparison of performance between the baseline model, training without selection, MLLR adaptation with the task-specific data and the proposed approach for selective training.

Table 5.7. Disk space and executing time required for selective training using sufficient statistics (SS)

Experiment	A, *ST_DelScan*	B, *ST_DelAdd*
Total Run Time	≈ 20 minutes	≈ 27 minutes
Total CPU Time	≈ 10 minutes	≈ 18 minutes
CPU Time Q-function	216 seconds	366 seconds
Model Size (ASCII)	1300 KB	1300 KB
Development data SS	368 KB	313 KB
Training data SS	400 KB	379 KB
Single utterance SS	78 KB	84 KB
Total disk space SS	2.5 GB	1.4 GB

the SS is 2.5 GB. Since the selection works utterance-based it is possible to reduce the additional space necessary to store the SS to zero at the cost of doubling computation time, if the *ST_DelScan* variant is used. However, the reduction of disk space is not a recommendable option for the *ST_DelAdd* variant.

It is clear, that building a task-dependent model with the proposed algorithm is feasible within a short period of time. Even if the model complexity and the size of the data pool increase, enough disk space can be provided easily and the additional computation time needed for utterance selection is only a fraction of the time necessary for one Baum-Welch training iteration.

5.1.2 Comparison of Acoustic Models

A monophone and a PTM acoustic model is built from scratch with all utterances in the corresponding data pool using HTK [31]. The monophone model consists of 3-state HMMs with up to 16 Gaussians densities (diagonal covariance matrix) per state. There is one HMM for each of the 40 phonemes in the standard

Table 5.8. The total number of physical HMMs, the number of distinct HMM states and the total number of parameters (means, covariances, weights and transition probabilities)

Nr.	AM Type	# phys. models	# states	# params
(A)	Monophone	43	129	103k
	PTM	765	785	210k
(B)	Monophone	43	129	103k
	PTM	572	628	200k

Table 5.9. Relationship between the number of Baum-Welch training iterations with the selected training data and the recognition performance (word accuracy in %)

Model Type	Training Iteration					
Monophone	init	1	2	3	5	8
(A) Preschool	46.9	50.2	49.7	50.2	**51.7**	**51.7**
(B) Elderly	73.6	75.1	**75.9**	75.1	75.0	74.7
PTM AM	init	1	2	3	5	8
(A) Preschool	53.0	55.5	**55.9**	**55.9**	55.7	55.3
(B) Elderly	76.7	77.9	77.5	77.7	77.7	**78.2**

Japanese phoneme set plus three silence HMMs (utterance begin, utterance end and short pause). Evaluation is also carried out for phonetic tied-mixture (PTM) models [45], which share one codebook of 32 Gaussians per state among state-clustered triphones with the center phone in common, but with mixture weights untied. Information about the complexity of each monophone and PTM acoustic model employed in experiments (A) and (B) is given in Table 5.8. PTM acoustic models enable fast decoding with the open-source LVCSR engine Julius [37] while maintaining a high recognition performance comparable to context-dependent triphone models. Other experimental conditions are given in Table 5.2.

For decoding the preschool children test set, a task-specific 4k word language model trained on transcriptions of preschool children utterances, and for decoding the elderly test set a 40k word language model trained on utterance transcriptions from the Takemaru database as well as texts from e-mails and Internet pages is employed.

Preschool-adapted Acoustic Model. The word accuracy of the initial monophone and PTM model built with all utterances in the data pool (containing only speech from elementary school children) is 46.9% and 53.0%, respectively. When applying selective training using 200 preschool utterances for likelihood computation, the accuracy increases up to 10% relative for the monophone (51.7%) and 5.5% relative for the PTM model (55.9%). 35% of the utterances in

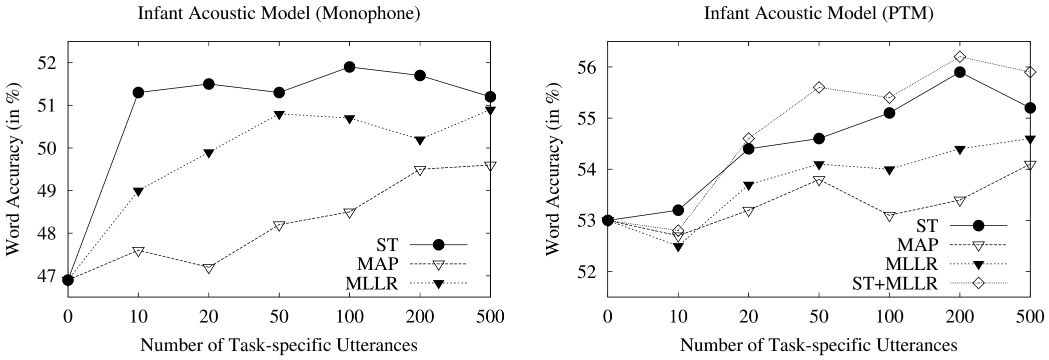

Figure 5.4. Influence of the amount of task-specific speech data on the performance of selective training (ST) and standard adaptation methods. Retraining with the selected data was conducted for three (monophone model) or eight (PTM model) iterations, respectively. MLLR adaptation is carried out for two, MAP adaptation for one iteration (Preschool Children Experiment A).

the data pool were selected.

Elderly-adapted Acoustic Model. The word accuracy of the initial monophone and PTM model built with all utterances in the data pool (containing only adult speech) is 73.6% and 76.7%, respectively. 53 utterances from elderly people are employed for likelihood-based utterance selection. There is a relative improvement of recognition accuracy of up to 3.1% for the monophone (75.9%) and up to 2.0% for the PTM model (78.2%). The selection rate of training utterances in the data pool was 44%.

Retraining with Selected Data. Table 5.9 shows the relationship between the number of Baum-Welch training iterations to train the initial acoustic model with the selected speech data and the recognition performance. Except for the context-independent monophone model for elderly people (peak after the second iteration), the recognition accuracy has the tendency to increase after several training iterations. Retraining of the initial acoustic model with the whole data pool did not improve the performance of the initial model.

Dependency on the Amount of Task-Specific Data. The performance in case of larger and smaller task-specific speech data sets for experiment (A) is depicted in Figure 5.4. It is clear that selective training is already effective with only 20 task-specific utterances. Maximum performance seems to be reached with about 100-200 utterances. Furthermore, it is apparent that selective training can provide a better model than standard adaptation methods such as MAP adaptation of means or MLLR adaptation of means and variances if there are only few task-specific data available. The combination of selective training and MLLR adaptation was not effective for the monophone model, but there were improvements for the PTM model.

Table 5.10 shows that the number of utterances selected from the data pool increases with the size of the task-specific data set, although not at the same rate.

Table 5.10. Relationship between the number of task-specific data and the number of utterances selected from the data pool

# Task-Specific	(A) Preschool Experiment	
Utterances	Monophone	PTM
10	8,715 (29%)	9,108 (31%)
20	9,426 (32%)	9,544 (32%)
50	9,609 (32%)	9,825 (33%)
100	10,300 (35%)	10,434 (35%)
200	10,252 (34%)	10,311 (35%)
500	10,852 (36%)	10,793 (36%)

Table 5.11. Performance (word accuracy) of high-cost models as more transcribed data collected with the Takemaru system (preschool speech) or from a separate database (elderly speech) becomes available

Model	Data Kind	# Data	Word Accuracy
Monophone	Preschool	2,000	49.9%
	(Takemaru)	3,000	50.4%
		5,000	51.5%
PTM	Preschool	3,000	54.9%
	(Takemaru)	5,000	55.5%
		10,000	56.8%
Monophone	Elderly	56,604	71.9%
PTM	(SJNAS)	56,604	73.9%

Even in case of only 20 utterances the selected training data suffice to train the initial acoustic model robustly.

Comparison with High-Cost Models. The experimental results so far showed the effectiveness and practical applicability of the proposed method to the problem of building a task-adapted acoustic model while using only a few task-specific speech data. However, it is not clear yet, how much more task-specific speech data have to be collected in order to achieve the same performance as with selective training. Table 5.11 shows the performance of models trained on either many thousand preschool utterances collected by *Takemaru* or the SJNAS corpus containing more than 50,000 utterances from about 300 different elderly persons (a database description can be found in [5]). The decision to use this speech corpus for comparison is due to the fact that only very few utterances from elderly people were collected by the dialogue system.

In case of experiment (A), if there are 10,000 transcribed preschool utterances available for retraining the initial model, a higher performance than with selective training can be achieved. Nevertheless, the difference in performance

Table 5.12. Summary of experimental results (word accuracy in %). Selective training (SelTrain), Training without selection (Initial), MLLR or MAP adaptation with task-specific data (Adapt), Performance with high cost acoustic models (HighCost)

Data → Model	Type	Baseline	Adapt.	SelTrain	HighCost
Takemaru DB	Monophone	46.9	50.7	**51.7**	51.5
Element → Preschool	PTM	53.0	54.4	**55.9**	55.5
Takemaru DB	Monophone	73.6	75.0	**75.9**	71.9
Adult → Elderly	PTM	76.7	77.5	**78.2**	73.9

between this well-trained (56.8%) and the initial model (46.9%) is reduced by 76% (relative). Furthermore, using only 5,000 preschool utterances for Baum-Welch training would not be enough to outperform selective training. In case of experiment (B), an acoustic model trained on a large database of elderly speech could not beat the performance of the initial acoustic model trained on adult speech collected with the Takemaru system. From this comparison it is clear that at least about 20-30 times more task-specific speech data would have to be collected in order to reach the performance of the model obtained with selective training.

Summary of Experimental Results A summary of experimental results is given in Figure 5.5 and Table 5.12. There are significant improvements in word accuracy over the initial model by employing selective training. Column "Adapt." shows the maximum performance among MAP and MLLR adaptation with the task-specific data. The difference in performance to selective training is large enough to be able to consider the proposed algorithm as a reasonable alternative for task-adaptation of acoustic models. Furthermore, the performance of AMs built by selective training is higher than high-cost AMs constructed from large amounts of task-specific data.

5.1.3 Conclusion

A framework for cost-effective task-adaptation of acoustic models using utterance-based selective training has been evaluated. It has been shown that it is possible to select relevant training utterances from a large data pool given only a small amount of task-specific development data. The method was applied to obtain a preschool-adapted and an elderly-adapted acoustic model.

The proposed training method is effective for context-independent monophone as well as context-dependent phonetically-tied mixture acoustic models. No definite conclusions could be drawn about which of the selective training algorithm variant is the better for recognition performance.

Selective training outperformed supervised adaptation with the task data.

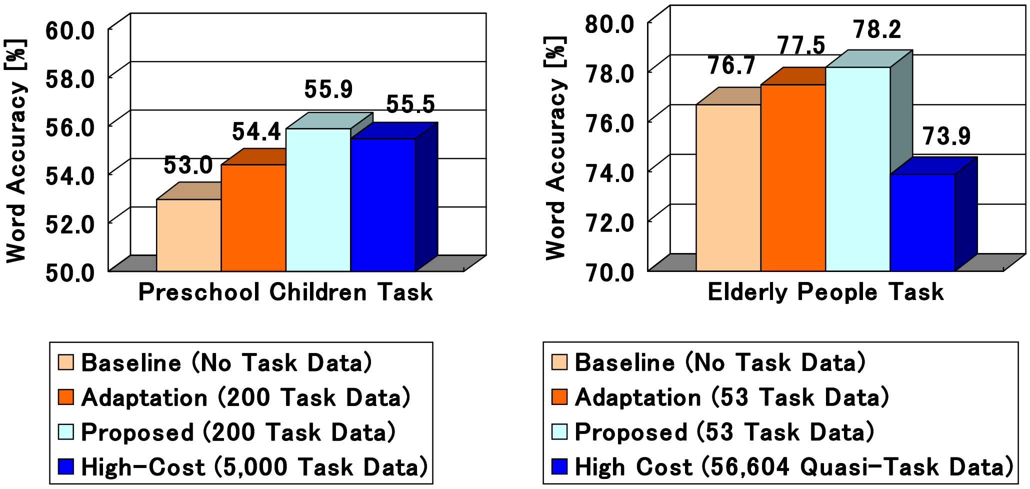

Figure 5.5. Comparison of baseline model, MLLR/MAP adaptation, proposed method for selective training and high-cost models considering ASR performance and amount of task-specific speech data.

The gap in performance to high-cost acoustic models constructed by supervised training using large amounts of task-specific data could be reduced up to 76% relative. Since the proposed method requires only moderate amounts of task-specific data, costs can be reduced drastically if a large pool of speech data similar to the task data is available.

The proposed method is also practically feasible. The additional time necessary for selective training is only a fraction of a conventional Baum-Welch training iteration. Providing the several GBs of disk space for storing the sufficient statistics is not problematic in current and future days.

5.2 Untranscribed Data Pool

Consider the case new speech data is collected for the development of a real-environment ASR application. To realize cost reduction only data collected during a restricted period after begin of system operation can be transcribed by humans. After that only unsupervised training with the unlabeled data can be carried out. However, unsupervised training requires a transcription of the unlabeled data. Automatic transcription by automatic speech recognition is always error-prone. The more transcription errors there are, the less effective will be unsupervised training. Furthermore, not all speech inputs collected should be employed for training, for example non-verbal inputs such as laughing or coughing, unintelligible inputs and inputs with strong background noise or multi-talk interference. Furthermore, it is often important to improve ASR performance by parallel decoding with multiple acoustic models. Therefore, it would be better to

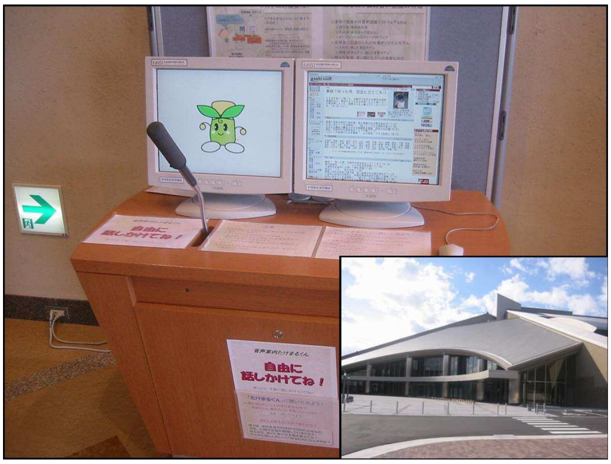

Figure 5.6. Speech-oriented guidance system *Takemaru* installed at the North Community Center in Ikoma City, Nara Prefecture, Japan.

employ only a subset of the unlabeled data to construct each acoustic model.

In the following the selective training algorithm is applied to build task-adapted acoustic models if the data pool is completely untranscribed. In experiments it is investigated in how far the proposed method can select training utterances with a high transcription accuracy from the desired speaker group and whether it can discard too noisy or low quality inputs. The effect of the period of speech data collection and transcription on speech recognition performance is also analyzed.

5.2.1 Target Application

As outlined in the introduction, employing one acoustic model for different applications and environments is difficult due to the task-dependency of speech recognition. In the following, acoustic model construction for a real-environment speech-oriented guidance system is considered as a realistic scenario for application development.

The purpose of a speech-oriented guidance system is to offer a certain group of users convenient access to proper information in a certain environment. While the information society is at the verge to an ubiquitous society, there is growing demand for this kind of services in any place. Although entering search queries via keyboard is still the prevailing method for accessing information, formulating one's question freely in natural language and using speech is a far more natural way to human-machine communication.

The *Takemaru* system [53] installed inside the entrance hall of the North community center in Ikoma city, Nara Prefecture, Japan is being operated since November 2002 (Figure 5.6). The indoor environment is relatively calm with a background noise level of approx. 50 dB(A). The place is frequently visited by adults and children, because it is a public facility with a library, a branch office

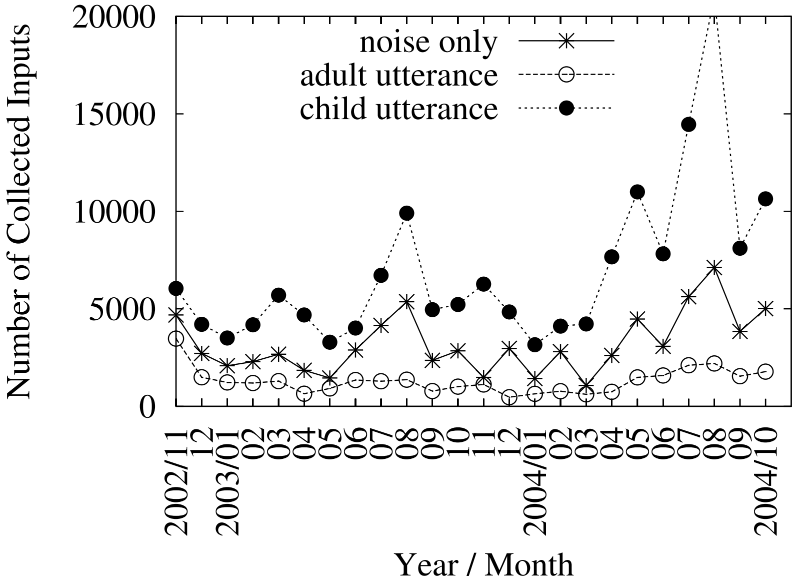

Figure 5.7. Inputs collected during the first two years of operation.

for residental services and there are weekly events. The system uses the mascot character of Ikoma city, *Takemaru*, as agent. The system can handle queries related to the agent, general information such as time, date, weather and news, the facility itself, surrounding area and sightseeing.

5.2.2 Problem Description

Various kinds of inputs are collected when deploying an ASR application in a real environment. There are speech and non-verbal inputs from several user groups as well as noise inputs from the surrounding environment. The data collected by *Takemaru* during the first two years is shown in Figure 5.7.

For system development, a large amount of the collected data is most often transcribed and labeled with tags (e.g. noise, validity, speaker group classification) by humans. This is a very costly and time-consuming process. If accurate transcriptions and speaker group labels are available for each utterance, it is straightforward to build high-quality speaker-group-dependent acoustic models. The purpose of building extra models for different speaker groups is to improve overall recognition performance by parallel decoding, and to optimize the performance separately for each speaker group. For example, the *Takemaru* system uses different models for recognizing and responding to inputs of adult and children users.

In order to reduce the costs of acoustic modeling, it is imperative to reduce the amount of data to be transcribed by humans. However, it is desirable that also the unlabeled data can be used effectively for model training. To achieve this, the employment of unsupervised training is inevitable.

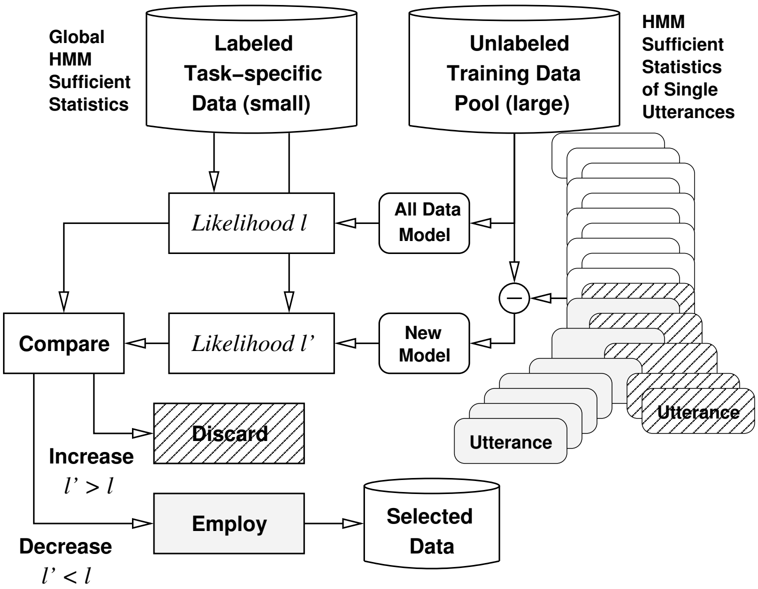

Figure 5.8. Selective training using a greedy maximum likelihood (ML) training utterance selection strategy.

5.2.3 Proposed Training Procedure

The proposed training procedure is a combination of unsupervised and selective training. Selective training is employed to alleviate the problems which arise when using unsupervised training.

Unsupervised Training. Unsupervised learning requires to transcribe the unlabeled data automatically. The initial model for automatic transcription can be built from scratch or by adapting an existing model with the labeled data. The automatically transcribed data can then be employed together with the labeled data to retrain the initial model. However, with introduction of unsupervised learning for real-environment data the following problems arise:

1. Automatic transcriptions are always error-prone

2. Speaker (group) of unlabeled inputs is unknown

3. Data kind is unknown (speech, noise, non-verbal)

Consequently, it is necessary to select an appropriate subset of the unlabeled data for acoustic modeling. Ideal training utterances are speech-only inputs, which have a high transcription label accuracy, do not contain strong noise interferences and belong to a certain speaker group.

Selective Training. It is possible to automatically select training utterances with these characteristics by applying our proposed method for selective training. A graphical illustration of the selective training algorithm is given in Figure 5.8. The starting point is a large training data pool $\mathcal{T}$ and a small development data

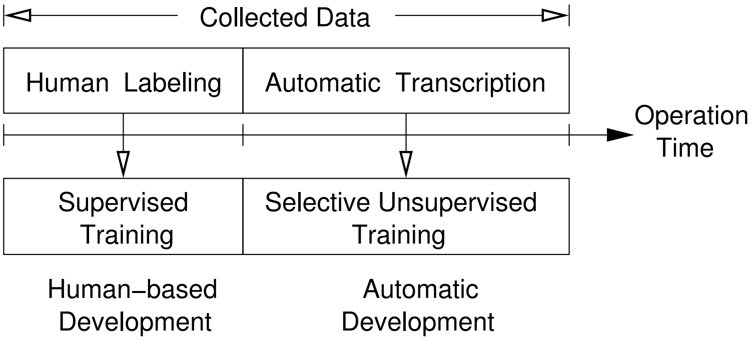

Figure 5.9. Framework for cost-effective acoustic model construction using unsupervised and selective training. After initial, short-term development follows long-term automatic development.

set $\mathcal{D}$. Here, the data pool consists of all unlabeled, collected inputs. Human-transcribed utterances from a certain speaker group form the development set. Let $\hat{\mathbf{\Theta}}$ denote the model parameters estimated on the selected data $\mathcal{S}$. The main idea of the proposed selection algorithm is to select a subset of utterances $\mathcal{S} \subseteq \mathcal{T}$ from the data pool so that the model likelihood $P(\mathcal{D}|\hat{\mathbf{\Theta}})$ given the development data is maximized. Since $\mathcal{D}$ consists of human-prepared example data, it can be expected that the algorithm selects high-quality speech utterances from the data pool matching the desired target task, e.g. speaker group.

5.2.4 Simulation Experiments

In the beginning, only a small amount of human-labeled data are employed for active system development. After that the system will retrain itself automatically with newly collected data. This is interesting for practical system development, since human efforts are only necessary at the beginning. The performance will improve over time without additional costs and human intervention.

In a simulation experiment, the influence of data collection and transcription period on speech recognition performance is analyzed. The less data are labeled by humans the lower the development costs but also the performance. Therefore, it is investigated how performance improves as more collected data for selective unsupervised training become available (Figure 5.9).

The procedure of acoustic model construction is illustrated in Figure 5.10. The main steps are as follows.

1. Retrain or adapt JNAS[35] model with the labeled data depending on the amount of available training data

2. Obtain initial model for automatic transcription and later retraining

3. Recognize all data in the unlabeled data pool

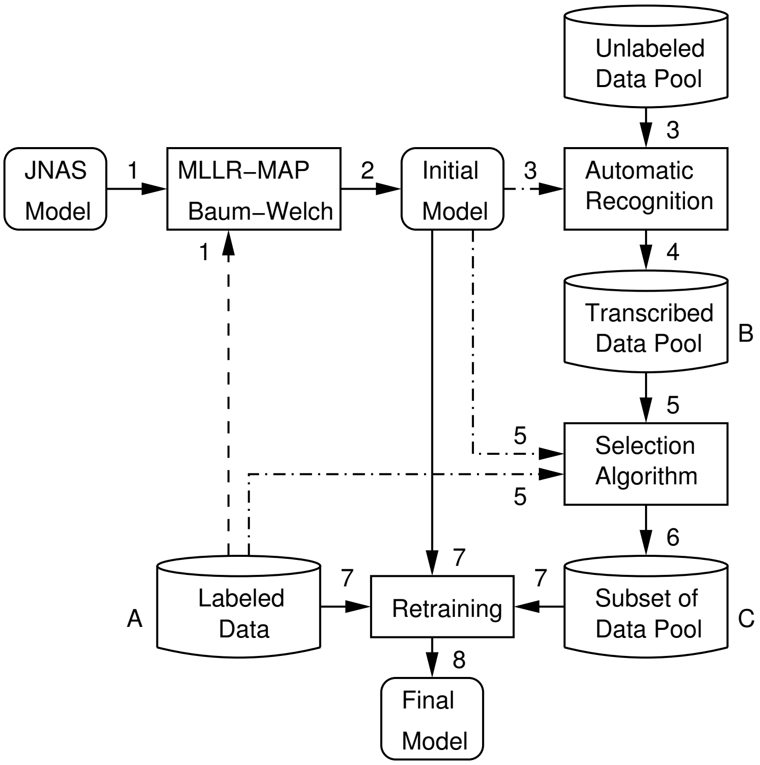

Figure 5.10. Procedure for acoustic model construction.

4. Obtain transcriptions for the unlabeled data
5. Selective training to maximize model likelihood given the labeled data
6. Obtain subset of the data pool
7. Retrain initial model with the labeled and selected utterances
8. Obtain the final acoustic model

Experimental conditions for acoustic model training, adaptation and evaluation, and the language model for automatic transcription are given in Table 5.13. When much training data is available, the acoustic model is trained using the Baum-Welch algorithm. Otherwise it is adapted in a two-step approach using MLLR-MAP: Firstly, mean vectors and diagonal covariance matrices of Gaussians are adapted using MLLR transforms [48]. After that MAP estimation [22] of mean vectors is carried out for the MLLR-transformed model.

Real-environment data collected with the *Takemaru* system are employed for the development simulation. The complete two-year *Takemaru* database is shown in Table 5.14. Training and evaluation data sets are given in Table 5.15. Evaluation data were selected randomly so that there is one utterance each for the most frequent 1,000 utterance transcriptions. The adult test set is gender-balanced

Table 5.13. Experimental conditions for selective, unsupervised training.

AM Training	HTK 3.2 [31]
Acoustic Model	PTM [45], 2,000 states, 8,256 Gaussians
Acoustic Features	12 MFCC, 12 Δ MFCC, Δ E
AM Training	Baum-Welch, 2 Iterations
AM Adaptation	MLLR-MAP, 256 Classes, 2 Iterations
Language Model	Takemaru, 3-gram, 42k vocabulary
ASR Engine	Julius 3.5 [37]

Table 5.14. Human-labeled part of the *Takemaru* database

Classification	rel. share	# inputs	Time [h]
Preschool Children	10.1%	27,535	14.3
Lower Grade School	39.0%	106,797	57.7
Higher Grade School	11.5%	31,402	15.8
Adults, Elderly	11.3%	31,100	14.1
Noise, Non-Verbals	28.1%	76,864	19.3
Total		273,698	121.2

with 1,000 utterances each from male and female speakers. The children test set consists of 1,000 utterances each for preschool, lower grade and higher grade school children. For supervised training or supervised adaptation only valid user inputs are employed, i.e. utterances with strong interfering noise and unintelligible inputs are discarded. Moreover, evaluation, labeled and unlabeled data are always selected to be mutually disjoint.

GMM-based Selection. A conventional method to select training data for a certain task is a pattern classification approach using statistical models such as Gaussian Mixture Models (GMM). Their application has already been successful, e.g. for speaker identification/verification [64] or the rejection of unusable inputs to a spoken dialogue system [46]. The disadvantage of the GMM-based approach over the proposed method for selective training is that it lacks an op-

Table 5.15. Evaluation data and maximum number of labeled data (for initial supervised training) and unlabeled data (for unsupervised learning) in the adult and child model experiment. Inputs shorter than 0.7 seconds have been discarded in advance

Experiment	Adult Model		Child Model	
Classification	# inputs	Time	# inputs	Time
Evaluation Data	2,000	65 min	3,000	95 min
Max Labeled	21,997	11 hrs	117,673	67 hrs
Max Unlabeled	238,920	115 hrs	237,920	115 hrs

timization criterion which is directly linked to the model quality. The proposed selective training method adopts the model likelihood given the task-specific data as optimization criterion.

Classification models are built for three classes: One GMM each is constructed from the labeled adult (and elderly), (preschool and school) child and noise (and non-verbal) data of each transcription period (see grouping in Table 5.14). The same 25-dimensional MFCC-based feature vector as for speech recognition is employed for constructing the GMM for each class from scratch. The number of mixture components is increased incrementally until 64 Gaussians are reached.

For automatic data selection, the unlabeled data of each corresponding data collection period are partitioned automatically into three classes. Data classified as noise are discarded completely. Data classified as adult (or child) are sorted by the GMM likelihood. The top-ranked data are employed together with the labeled data for building the adult and child acoustic model, respectively. For a fair comparison with respect to the number of training data the same selection rate as determined by the proposed selective training algorithm is used.

5.2.5 Experimental Results

The *proposed method*, which employs labeled and unlabeled selected data (A+C) for training, is compared with *supervised training* when using only labeled data (A) and *semi-supervised training* when using labeled and unlabeled data (A+B). Furthermore, it is analyzed whether the selective training algorithm actually selects speech data of the expected characteristics from the unlabeled data pool.

Comparison of Training Methods The case of a fixed and short transcription period of only one week is considered first. The performance is shown in Figure 5.11 for adults and children, respectively. It can be observed that the performance with semi-supervised training increases for children but decreases for adult speakers. This is due to the fact that most of the collected data are from children users. Consequently, performance improves remarkably for adult speakers when using the proposed method. For children there are slight improvements for short as well as long data collection periods. It is clear that the proposed method outperforms both supervised and semi-supervised training.

The influence of more human transcriptions on performance after 18 months (fixed) of data collection is shown in Figure 5.12 The more human-transcribed data is available for training, the higher is the overall performance. The difference between the proposed method and supervised training becomes small for both adult and children after three months. This shows that the proposed method is effective especially when only very few data can be transcribed by humans. Moreover, it is clear that the conventional GMM-based method does not outperform the proposed method in selecting arbitrary task-specific training data. While the GMM-based method is effective in selecting adult utterances with a

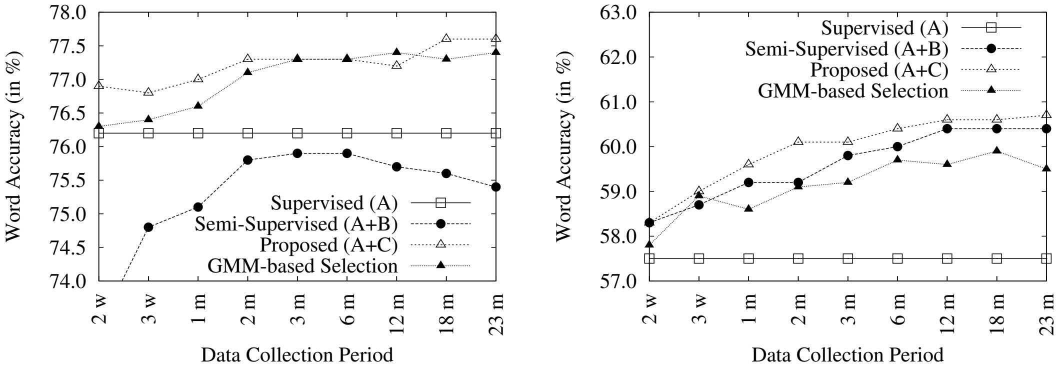

Figure 5.11. Performance for adult and child AM (one week transcription).

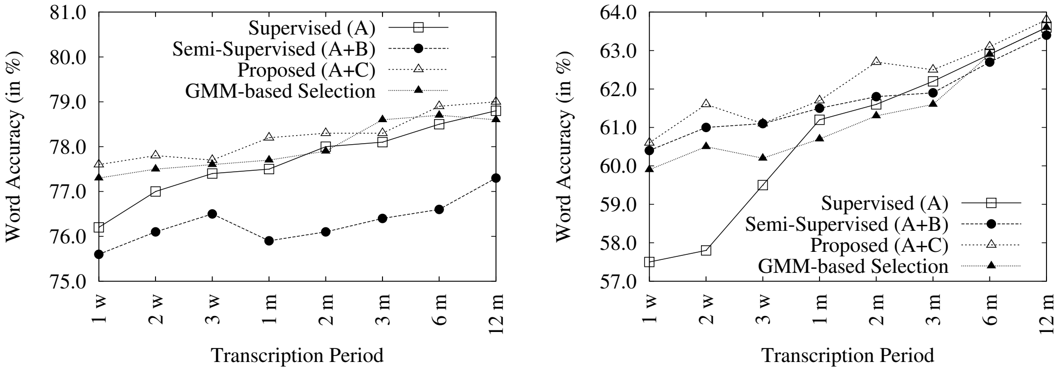

Figure 5.12. Performance for adult and child AM (18 month data collection).

performance close to selective training, its overall performance is lower than both semi-supervised training and the proposed method for building the child model.

The number of labeled, unlabeled pool and unlabeled selected data used to train each corresponding model is given in Table 5.16. The sum of the number of labeled and unlabeled pool data is not equal for each column, because noise data, non-verbal data, unintelligible data and utterances except from the target speaker group are excluded from the labeled data. Selective training can augment the existing labeled data with an appropriate subset of the unlabeled data. This avoids data insufficiency for model training and improves performance especially when the data transcription period is short.

Validity of Data Selection The selected data of the one week transcription and 18 months data collection experiment are analyzed with respect to the human labels and speech recognition accuracy on utterance basis.

A breakdown with respect to human-assigned age group and noise labels is shown in Table 5.17. It is clear that the proposed selective training algorithm is able to reduce the relative share of noise data and also increases the relative share of utterances from the desired speaker group. This means that the second and

Table 5.16. Number of labeled (A), unlabeled pool (B) and unlabeled selected data (C) for each transcription period (18 months data collection). The time units for collection and transcription periods are weeks (w) and months (m)

Child Exp.	1 w	2 w	1 m	2 m	3 m	6 m	12 m
# labeled	2k	3k	6k	8k	11k	23k	50k
# pool	139k	135k	127k	123k	117k	94k	40k
# select	34k	34k	32k	30k	30k	22k	13k
% select	24.4	25.5	24.8	24.3	25.8	23.8	32.7
Adult Exp.	**1 w**	**2 w**	**1 m**	**2 m**	**3 m**	**6 m**	**12 m**
# labeled	1k	2k	4k	4k	5k	8k	13k
# pool	140k	136k	128k	124k	117k	95k	41k
# select	8k	11k	9k	9k	9k	8k	5k
% select	5.4	7.8	7.0	7.4	7.7	8.6	12.2

Table 5.17. Breakdown with respect to human-assigned labels of the pool data before selection and the subset selected with the selective training algorithm

Experiment	Adult Model	Child Model
Data Pool → Selected	140k → 8k	139k → 34k
Adult Inputs	12.0% → 73.9%	12.3% → 8.4%
Child Inputs	64.4% → 14.7%	63.0% → 78.4%
Noise Inputs	23.6% → 11.4%	23.7% → 13.1%

third initially mentioned problems arising from the employment of unsupervised learning are already alleviated by the proposed method.

Selected utterances had more often a relatively high, discarded utterances a relatively low recognition accuracy. When building the adult acoustic model, the average word accuracy (correct rate) of selected and discarded speech inputs is 78.6% (81.7%) and 41.9% (48.5%), respectively. For the children experiments the rates are 57.2% (63.4%) and 53.0% (59.5%), respectively. This shows that utterances with an erroneous transcriptions are more likely to be discarded so that the first initially mentioned problem of unsupervised learning is also addressed.

Development Cost Reduction From the experimental results in Figure 5.12 it can be verified that the proposed method is effective for reduction of development costs in comparison to supervised training. Comparing supervised learning for a transcription period of two months and the proposed method in case of a transcription period of about two weeks, the number of human-transcribed data employed for model retraining is reduced more than half. At the same time the recognition performance is almost equal. The same can be observed when comparing six months supervised learning with the proposed method in case of two months transcription. Assuming the same cost factors as defined in Section 2.5.4,

Table 5.18. Comparison of supervised learning and proposed method regarding transcription period and number of human-transcribed data employed for model training

Learning Method	Supervised	Proposed	Supervised	Proposed
Transcription period	2 months	0.5 months	6 months	2 months
# transcribed data	21,037	8,728	49,972	21,037
# collected data	21,037	145,935	49,972	145,935
Child/Adult WA [%]	61.6 / 78.0	61.6 / 77.8	62.7 / 78.5	62.9 / 78.3
Costs [points]	625,925	455,407	1,349,300	750,823
Cost Reduction	-	37%	-	44%

the costs for system construction depending on each approach can be calculated. It is clear from Figure 5.18 that the proposed method is effective in reducing the development costs by up to 44% (relative) depending on the considered transcription period for supervised learning.

5.2.6 Conclusion

A powerful combination of unsupervised and selective training has been proposed and evaluated to reduce the costs of acoustic modeling for real-environment speech-based applications. The employment of unsupervised learning is inevitable to avoid the very costly and time-consuming process of transcribing large amounts of speech data by humans as much as possible. The purpose of applying selective training is to alleviate the problems which arise when using unsupervised training. The idea to select additional training utterances from an unlabeled data pool, so that the model likelihood given the labeled data increases, is promising. Experimental results show that the proposed selective training algorithm can select automatically utterances of the desired speaker group, e.g. adult or child, and discard most of the noise-only inputs as well as data with a lower recognition accuracy. From analyzing the influence of the data collection and transcription period on the recognition performance it was clear that the proposed method is especially effective when the amount of data labeled by humans is restricted.

Chapter 6

Summary

It has been outlined in Chapter 1, that task-dependency and the costs for speech data collection and human transcription are major difficulties when developing real-environment ASR applications. Several methods for reducing development costs are known from literature. Reuse of existing data, unsupervised learning, active learning and lightly supervised learning are the most prominent examples. The effect of reusing existing data and models is investigated for real-environment speech-oriented guidance systems in this work. However, mere data reuse and conventional learning methods have the drawback that the issue of task-dependency is not addressed sufficiently. Therefore, a cost-effective approach for selective training to construct task-adapted models is proposed in this work.

6.1 Data, Model and System Reuse

Development and long-term operation of the speech-oriented guidance system *Takemaru* in a real environment has been investigated in Section 3.3. *Takemaru* is installed at the local community center. More than 1.2 million speech inputs have been collected during five years. Collection of real user data is required because ASR task and Q&A domain of a guidance system are defined by the target environment and potential users. The purpose of the development simulation is to determine empirically the amount of real-environment data which have to be prepared to build a system with reasonable performance.

Although depending on modeling capacities and domain complexity in general, experimental results showed that performance saturates with 10-15k training utterances for the acoustic model, but 40-50k training utterances for the language model and 40k-50k human-labeled Q&A pairs for compiling the question and answer database. Efforts regarding the Q&A database were most important to improve the system's response accuracy.

However, human transcription of such large amounts of speech data is very costly. Therefore, the effect of reusing the well-trained *Takemaru* prototype system to build the *Kita* systems for a local train station was investigated in Section 3.4. *Takemaru* reuse, *Takemaru* update and from-scratch development with varying amounts of task-specific data from the *Kita* environment were evaluated. From the experimental results it was clear that the *Takemaru* ASR module is highly portable to the *Kita* environment. ASR performance already showed signs of saturation after short-term update.

On the other hand, the reusability of *Takemaru*'s Q&A module was relatively low, because response accuracy was poor before updating the Q&A module. Nevertheless, there were remarkable improvements after adding Q&A pairs collected in the target environment. This implicates that it will always be important to take the behavior of actual users under real conditions into account to build a system with high user satisfaction.

By reusing *Takemaru* data and models, response performance is improved by 2.1% - 3.7% absolute. Moreover, it was possible to reduce the system development period more than half (3 → 1 month, 5 → 2 months) and costs for data collection and preparation by more than 40% without compromising the performance.

6.2 Selective Training for Task Adaptation

To obtain a robust acoustic model for a certain speech recognition task, a large amount of training speech data is necessary. However, the preparation of speech data including recording and transcription is very costly and time consuming. Although there are attempts to build generic acoustic models which are portable among different applications, speech recognition performance is typically task-dependent. Therefore, a method for automatically building task-dependent acoustic models based on selective training has been proposed in Chapter 4.

Instead of preparing a new large speech database whenever building a new system, only a small amount of task-specific development data need to be collected. Based on the target model likelihood given the development data, utterances similar to the development data are selected from a data pool of existing speech data resources. Since there are too many possibilities for selecting a data subset from a larger database in general, a greedy selection strategy has to be employed. The proposed selective training algorithm either deletes single utterances temporarily or alternates between successive deletion and addition of multiple utterances. In order to make selective training computationally practical, model retraining and likelihood calculation need to be fast. It is shown, that the model likelihood can be calculated instantaneously based on sufficient statistics without the need for explicit reconstruction of model parameters.

Selective training was evaluated in Section 5.1 for human-transcribed data. It is rather difficult to collect large amounts of speech data from preschool children and elderly people. Furthermore, the ASR performance for preschool children is poor with children and adult acoustic models. Therefore, the idea was to construct a preschool and elderly-adapted acoustic model by selecting training utterances from school children and adult speech data, respectively. A relative improvement in word accuracy of up to 10% over training with all available training data was achieved. Furthermore, the performance of selective training was higher than MLLR or MAP adaptation with the task-specific development data. Synergetic effects could be observed by combining selective training and conventional task adaptation methods.

6.3 Selective Unsupervised AM Training

Development of an ASR application such as a speech-oriented guidance system for a real environment is expensive. Most of the costs are due to human labeling of newly collected speech data to construct the acoustic model for speech recognition. Employment of existing models or sharing models across multiple applications is often difficult, because the characteristics of speech depend on various factors such as possible users, their speaking style and the acoustic environment. Therefore, a combination of unsupervised learning and selective training to reduce the development costs has been investigated in Section 5.2.

The employment of unsupervised learning alone is problematic due to the task-dependency of speech recognition and because automatic transcription of speech is error-prone. A theoretically well-defined approach to automatic selection of high quality and task-specific speech data from an unlabeled data pool is applied. Only those unlabeled data which increase the model likelihood given the labeled high-quality data are actually employed for unsupervised training.

The effectivity of the proposed method has been investigated by a simulation experiment to construct adult and child acoustic models for a speech-oriented guidance system. A two-year real-environment speech database was employed for the development simulation. Experimental results show that the employment of selective training alleviates the problems of unsupervised learning. The proposed method selected speech utterances of a certain speaker group but discarded noise inputs and utterances with lower recognition rate.

The simulation experiment was carried out for several combinations of collection and human transcription periods. It was found empirically that the proposed method is especially effective if only relatively few data can be labeled and transcribed by humans. There was an overall improvement in performance over supervised training for each transcription period if unlabeled data from a longer data collection period are available. The same level of performance could be reached with less human transcribed data. Development costs for data collection and human transcription were reduced by up to 40% without compromising the performance.

6.4 Outlook and Future Work

Although a wide range of topics and tasks have been considered in this work, there is still a number of issues worth to be considered.

Applications of Selective Training As already mentioned in Section 4.3, selective training could be employed for cross-language acoustic modeling to reduce the development costs of ASR systems for rare languages, or languages with sparse speech data resources. However, multi-lingual language resources to investigate this possibility have not been available for this work. It remains to conduct experiments for acoustic model construction for a certain language using speech data from other languages.

Employ Selective Training for Active Learning Selective training has been applied to an untranscribed data pool in Section 5.2 to avoid the costs for human transcriptions. Since performance with automatically transcribed speech data is limited, it is worth considering to employ selective training to select those data to be transcribed by humans similar to active learning.

Combination of Active Learning and Selective Training Furthermore, it should also be considered how the data requirements for development and adaptation of a real-environment guidance systems as conducted in Sections 3.3 and 3.4 changes when active learning is incorporated.

General Framework for Selective Training In this work, selective training has been applied to the problem of acoustic modeling for speech recognition. Nevertheless, the selective training framework can in principle be applied to any problem which is based on statistical models having sufficient statistics. Therefore, it is worth to consider selective training also for other pattern classification problems. For example, the selective use of text data for language modeling [69] and leave-one-out optimization of the question and answer database of a dialogue system [74] have been investigated recently.

Appendix A

Experimental Results

A.1 Takemaru Development Simulation

Table A.1. Results for *Takemaru* development simulation using adult data. The first column shows the end of each development period, excluding validation data, test data and the data of the corresponding month

Period	Collected	Employed	LM			WA	Q&A DB	RA
(until)	(# data)	(# data)	OOV	PP	Vocab.	[%]	(# pairs)	[%]
2003/01	1,489	1,310	8.2	20.7	609	68.9	601	52.9
2003/02	2,713	2,379	5.8	15.8	870	73.3	1,025	61.2
2003/03	3,907	3,401	4.4	14.5	1,156	75.2	1,413	62.8
2003/04	5,202	4,466	3.8	13.7	1,369	75.7	1,775	65.3
2003/05	5,848	4,981	3.2	12.6	1,457	76.6	1,928	67.0
2003/06	6,766	5,606	3.0	12.0	1,604	77.4	2,179	67.8
2003/07	8,113	6,647	2.6	11.3	1,759	78.3	2,490	69.2
2003/08	9,401	7,519	2.4	10.9	1,941	77.9	2,770	69.6
2003/10	10,183	8,060	2.3	10.8	2,003	78.3	2,910	70.1
2003/11	11,195	8,779	2.3	10.6	2,097	78.2	3,068	70.9
2003/12	12,313	9,470	2.2	10.5	2,222	78.5	3,262	71.0
2004/01	12,782	9,778	2.1	10.4	2,264	78.5	3,342	71.0
2004/02	13,425	10,332	2.0	10.4	2,327	80.0	3,471	70.8
2004/03	14,202	10,957	2.0	10.2	2,406	78.6	3,631	71.7
2004/04	14,818	11,372	1.7	10.2	2,462	78.8	3,721	71.7
2004/05	15,564	11,815	1.7	10.2	2,534	78.8	3,752	71.1
2004/06	17,045	12,640	1.7	10.1	2,647	78.7	3,794	71.4
2004/07	18,628	13,443	1.7	10.1	2,759	78.5	3,841	71.4
2004/08	20,734	14,321	1.6	10.0	2,883	78.7	3,878	71.5
2004/09	22,932	15,100	1.6	9.9	2,983	79.3	3,893	71.5
2004/10	24,480	15,664	1.6	9.9	3,054	79.4	3,906	71.9
2004/11	26,259	16,305	1.6	9.9	3,110	79.5	4,048	72.1

Tables A.1 and A.2 show the detailed result of the development simulation for the *Takemaru* system. The training period starts from December 1st, 2002. Valid data collected in August 2003 are employed for evaluation of speech recognition and response performance. The interpolation weight for merging the all data with the speaker group (adult or child) language model is determined using the validation data (November 2002). Statistics of collected and employed data are based on the *Takemaru* ACCESS database of labels and transcriptions, which has been constructed by the technical assistants of Acoustics and Speech Processing Laboratory. Only user utterances with meaningful contents and without noise tags have been considered as usable and were employed for model training and evaluation. The perplexity (PP) has been calculated using a tri-gram language model with a fixed (i.e. two year) vocabulary.

Table A.2. Results for Takemaru Development Simulation (Child Data)

Period	Collected	Employed	LM			WA	Q&A DB	RA
(until)	(# data)	(# data)	OOV	PP	Vocab.	[%]	(# pairs)	[%]
2003/01	4,207	2,833	10.3	35.9	1,194	52.1	1,319	43.0
2003/02	7,710	5,073	7.6	29.1	1,710	56.1	2,117	47.5
2003/03	11,888	8,023	5.8	25.7	2,291	58.5	3,224	49.8
2003/04	17,592	12,120	5.0	24.0	2,884	59.7	4,566	51.8
2003/05	22,277	15,381	4.4	22.4	3,288	60.9	5,559	54.1
2003/06	25,566	17,405	4.1	21.6	3,568	60.8	6,205	53.7
2003/07	29,580	20,035	3.7	20.6	3,854	61.2	7,086	54.2
2003/08	36,299	24,301	3.2	19.2	4,423	61.0	8,807	54.2
2003/10	41,256	27,454	3.1	18.8	4,760	61.5	9,855	54.9
2003/11	46,477	30,517	2.8	18.4	5,069	61.8	10,847	55.1
2003/12	52,744	34,377	2.5	18.0	5,538	61.7	12,327	55.6
2004/01	57,585	37,211	2.4	17.8	5,808	61.9	13,220	56.1
2004/02	60,744	39,568	2.3	17.6	6,022	61.8	13,910	56.2
2004/03	64,857	42,384	2.2	17.4	6,263	62.0	14,840	56.5
2004/04	69,082	45,233	2.1	17.2	6,527	61.9	15,881	56.9
2004/05	76,750	49,169	2.1	17.0	6,864	61.8	16,142	56.7
2004/06	87,748	54,140	2.0	16.9	7,248	62.2	16,416	56.6
2004/07	95,567	57,179	1.9	16.8	7,424	61.9	16,562	56.4
2004/08	110,027	62,752	1.8	16.6	7,798	62.1	16,777	56.0
2004/09	131,036	69,866	1.7	16.5	8,234	61.9	16,998	55.9
2004/10	139,143	72,293	1.7	16.5	8,343	61.9	17,062	55.9
2004/11	149,782	75,082	1.7	16.5	8,486	62.0	17,863	55.8

A.2 Kita-chan and Kita-chan Robot Adaptation

Tables A.3 and A.4 show the detailed result of the development simulation for the *Kita* systems. The acoustic and language model of the two-year *Takemaru* prototype system were reused and adapted with *Kita* data. The initial Q&A database contains human-edited *Takemaru* data. The adaptation period starts April 1st, 2006 and ends January 31st, 2007. However only data collected in April, June, July, August, September 2006 and January 2007 have been employed. Valid data collected in May 2006 are employed for evaluation of speech recognition and response performance. The interpolation weight for merging the all data with the speaker group (adult or child) language model is determined using the validation data (also May 2006, but different days than evaluation data). Statistics of collected and employed data are based on the *Kitachan* ACCESS database of labels and transcriptions, which has been constructed by the technical assistants of Acoustics and Speech Processing Laboratory. Only user utterances with meaningful contents and without noise tags have been considered as usable and were employed for model training and evaluation. The perplexity (PP) has been calculated using a tri-gram language model with a fixed (i.e. two year *Takemaru* plus six months *Kita*) vocabulary.

Table A.3. Results for Kita-chan and Kita-chan Robot Adaptation (Adult Data)

Period	Collected	Employed	LM			WA	Q&A DB	RA
(until)	(# data)	(# data)	OOV	PP	Vocab.	[%]	(# pairs)	[%]
Takemaru	-	-	2.9	17.5	3,785	74.3	2,761	52.0
2006/05	6,872	4,932	1.8	15.9	4,307	77.0	4,914	67.2
2006/07	9,569	6,547	1.6	15.3	4,482	77.9	5,534	69.9
2006/08	11,814	8,066	1.6	14.9	4,596	78.3	5,987	71.0
2006/09	13,838	9,472	1.5	14.7	4,682	78.1	6,471	70.0
2006/10	15,378	10,570	1.5	14.5	4,751	78.4	6,778	70.6
2007/02	18,191	11,276	1.5	14.5	4,822	78.7	7,018	70.8

Table A.4. Results for Kita-chan and Kita-chan Robot Adaptation (Child Data)

Period	Collected	Employed	LM			WA	Q&A DB	RA
(until)	(# data)	(# data)	OOV	PP	Vocab.	[%]	(# pairs)	[%]
Takemaru	-	-	2.4	25.8	10,344	57.0	5,062	45.5
2006/05	9,653	6,711	1.9	25.1	10,704	59.9	8,350	54.7
2006/07	14,897	9,809	1.8	24.5	10,846	59.8	9,693	55.6
2006/08	19,675	12,785	1.8	24.2	10,954	60.1	10,742	56.7
2006/09	24,951	15,691	1.7	24.0	11,053	60.1	11,929	57.8
2006/10	28,509	17,808	1.7	24.0	11,133	60.3	12,664	58.1
2007/02	33,460	18,720	1.7	23.9	11,174	60.4	13,018	58.1

A.3 Selective Unsupervised Training

The evaluation of selective, unsupervised training has been carried out in Section 5.2. Only the result for selected data collection and data transcription periods have been shown. The complete results of the development simulation are given in Figures A.2 and A.1. The performance of combinations of selected transcription (vertical axis) and collection (horizontal axis) periods are shown. The overall improvement through the proposed method is clear by comparing the two graphs in each figure. It is possible to identify pairs of collection and transcription periods with equal performance. For example, by transcribing only two weeks of data collected during 12 months the same performance as when transcribing the data from two months can be obtained.

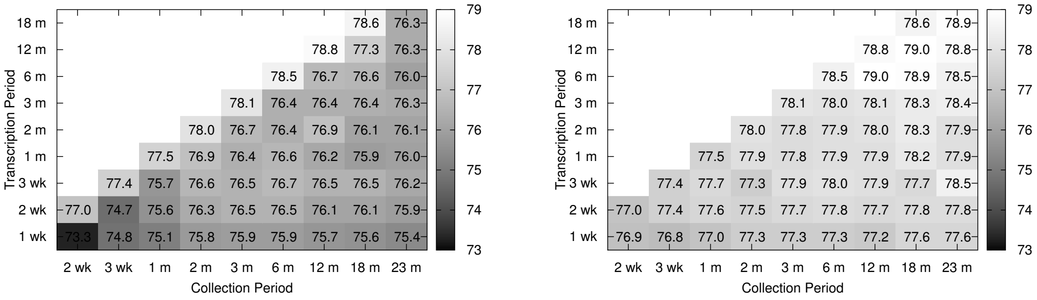

Figure A.1. Left: Training with labeled and unlabeled pool data. Right: Training with labeled and selected unlabeled data (Adults).

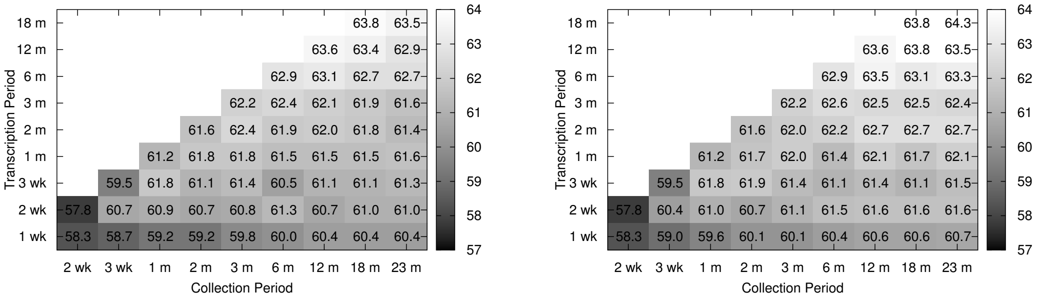

Figure A.2. Left: Training with labeled and unlabeled pool data. Right: Training with labeled and selected unlabeled data (Children).

Appendix B

Glossary

AM	Acoustic Model
ATR	Advanced Telecommunications Research, International
ASR	Automatic Speech Recognition
BN	Broadcast News
CALL	Computer Assisted Language Learning
CLI	Command Line Interface
COR	Correlation
DFT	Discrete Fourier Transform
DP	Dynamic Programming
DTW	Dynamic Time Warping
EM	Expectation Maximization
GMM	Gaussian Mixture Model
GPS	Global Positioning System
GUI	Graphical User Interface
HMM	Hidden Markov Model
HTK	Hidden Markov Model Toolkit
IBM	International Business Machines
JNAS	Japanese Newspaper Article Sentences (Adult Speech Database)
Takemaru	Speech-oriented Guidance System *Takemaru-kun*
Kita	Speech-oriented Guidance Systems *Kita-chan* and *Kita-robo*
KLD	Kullback-Leibler Divergence (relative entropy)
LM	Language Model
LPC	Linear Predictive Coding
LVCSR	Large Vocabulary Continuous Speech Recognition
MAP	Maximum A Posteriori
MFCC	Mel-Frequency Cepstrum Coefficients
ML	Maximum Likelihood
MLLR	Maximum Likelihood Linear Regression
MMI	Maximum Mutual Information
NUI	Natural User Interface
OOV	Out-Of-Vocabulary

PCA	Principal Component Analysis
PLP	Perceptual Linear Prediction
PP	Perplexity
PTM	Phonetically-Tied Mixture Model
QADB	Question and Answer Database
RA	Response Accuracy
SJNAS	Senior JNAS (Senior Speech Database)
SS	Sufficient Statistics
ST	Selective Training
ST_DelScan	Delete Scan Selective Training Algorithm
ST_DelAdd	Deletion/Addition Selective Training Algorithm
TUI	Text(ual) User Interface
VAD	Voice Activity Detection
WA	Word Accuracy
WSJ	Wall Street Journal

References

[1] M. Akbacak, Y. Gao, L. Gu, and H.-K. J. Kuo. Rapid Transition to New Spoken Dialogue Domains: Language Model Training using Knowledge from Previous Domain Applications and Web Text Resources. In *Proc. of Interspeech*, pages 1873–1876, 2005.

[2] Advanced Media, Inc.
`http://www.advanced-media.co.jp/english/`.

[3] T. Anastasakos, J. McDonough, R. Schwartz, and J. Makhoul. A Compact Model for Speaker-Adaptive Training. In *Proceedings of the International Conference on Spoken Language Processing*, pages 1137–1140, 1996.

[4] L. M. Arslan and J. H. L. Hansen. Selective Training in Hidden Markov Model Recognition. *IEEE Transactions on Speech and Audio Processing*, 7(1):46–54, 1999.

[5] A. Baba, S. Yoshizawa, M. Yamada, A. Lee, and K. Shikano. Elderly acoustic model for large vocabulary continuous speech recognition. In *European Conference on Speech Communication and Technology*, pages 1657–1660, 2001.

[6] L. Baum. An Inequality and Associated Maximization Technique in Statistical Estimation of Probabilistic Functions of a Markov Process. *Inequalities*, 3:1–8, 1972.

[7] M. Berouti, R. Schwartz, and J. Makhoul. Enhancement of Speech Corrupted by Acoustic Noise. In *Proc. of ICASSP*, pages 208–211, 1979.

[8] S.F. Boll. Suppression of Acoustic Noise in Speech using Spectral Subtraction. *IEEE Transactions on Acoustics, Speech and Signal Processing*, 27(2):113–120, 1979.

[9] I. N. Bronstein and K. A. Semendjajew. *Taschenbuch der Mathematik.* Harri Deutsch, Frankfurt am Main, Thun, 1997.

[10] Morphological Parser for the Japanese Language
`http://chasen-legacy.sourceforge.jp/`.

[11] L. Chen, L. Lamel, and J.-L. Gauvain. Lightly Supervised Acoustic Model Training using Consensus Networks. In *Proc. of ICASSP*, pages 189–192, 2004.

[12] S.F. Chen and J. Goodman. An Empirical Study of Smoothing Techniques for Language Modeling. *Computer, Speech and Language*, 13(4):359–394, 1999.

[13] T. Cincarek, R. Gruhn, and S. Nakamura. Speech Recognition for Multiple Non-Native Accent Groups with Speaker-Group-Dependent Acoustic Models. In *Proc. of ICSLP*, pages 1509–1512, 2004.

[14] T. Cincarek, I. Shindo, T. Toda, H. Saruwatari, and K. Shikano. Development of Preschool Children Subsystem for ASR and Q&A in a Real-Environment Speech-Oriented Guidance Task. In *Proc. of Interspeech*, pages 1469–1472, 2007.

[15] S.K. Das and M.A. Picheny. *Issues in Practical Large Vocabulary Isolated Word Recognition: The IBM Tangora System.* Kluwer, Boston, 1996.

[16] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum Likelihood from Incomplete Data via the EM Algorithm. *J.R. Statistical Society*, 1(39):1–38, 1977.

[17] V. Diakoloukas, V. Digalakis, L. Neumeyer, and J. Kaja. Development of Dialect-Specific Speech Recognizers Using Adaptation Methods. In *Proc. of ICASSP*, volume 2, pages 1455–1458, 1997.

[18] R. O. Duda, P. E. Hart, and D. G. Stork. *Pattern Classification.* John Wiley & Sons, Inc., 2001.

[19] G.W. Elko. Microphone Array Systems for Hands-Free Telecommunication. In *Speech Communication*, volume 20, pages 229–240, 1996.

[20] G.D. Fabbrizio, G. Tur, and D. Hakkani-Tür. Bootstrapping Spoken Dialogue Systems with Data Reuse. In *5th Workshop on Discourse and Dialogue (SIGdial)*, pages 72–80, 2004.

[21] Y. Gao, L. Gu, and H.-K. J. Kuo. Portability Challenges in Developing Interactive Dialogue Systems. In *International Conference on Acoustics, Speech, and Signal Processing*, pages 1017–1020, 2005.

[22] J.-L. Gauvain and C.-H. Lee. Maximum a Posteriori Estimation for Multivariate Gaussian Mixture Observations of Markov Chains. *IEEE Transactions on Speech and Audio Processing*, 2(2):291–298, 1994.

[23] D. Giuliani and M. Federico. Unsupervised language and acoustic model adaptation for cross-domain portability. In *ISCA ITRW Adaptation Methods for Speech Recognition*, pages 183–186, 2001.

[24] B. Gold and N. Morgan. *Speech and Audio Signal Processing.* John Wiley & Sons, Inc., 605 Third Avenue, New York, USA, 2000.

[25] R. Gomez, A. Lee, T. Toda, H. Saruwatari, and K. Shikano. Improving Rapid Unsupervised Speaker Adaptation Based on HMM Sufficient Statistics in Noisy Environments using Multi-Template Models. *IEICE Trans. on Information and Systems*, E89-D(3):998–1005, 2006.

[26] A.L. Gorin, G. Riccardi, and J.H. Wright. How may i help you? *Speech Communication*, 23(1/2):113–127, 1997.

[27] N. Gupta, G. Tur, D. Hakkani-Tür, S. Bangalore, G. Riccardi, and M. Gilbert. The AT&T Spoken Language Understanding System. *IEEE Trans. on Audio Speech and Language Processing*, 14(1):213–222, 2006.

[28] D. Hakkani-Tür, G. Riccardi, and A. Gorin. Active Learning for Automatic Speech Recognition. In *International Conference on Acoustics, Speech, and Signal Processing*, volume 4, pages 3904–3907, 2002.

[29] P. Heracleous, Y. Nakajima, A. Lee, H. Saruwatari, and K. Shikano. Audible (Normal) Speech and Inaudible Murmur Recognition Using NAM Microphone. In *Proc. of EUSIPCO*, pages 329–332, 2004.

[30] T. Hori, C. Hori, and Y. Minami. Fast On-The-Fly Composition for Weighted Finite-State Transducers in 1.8 Million-Word Vocabulary Continuous Speech Recognition. In *Proc. of Interspeech*, pages 859–862, 2004.

[31] HTK Speech Recognition Toolkit
`http://htk.eng.cam.ac.uk/`.

[32] C. Huang, T. Chen, and E. Chang. Transformation and Combination of Hidden Markov Models for Speaker Selection Training. In *Proceedings of the International Conference on Spoken Language Processing*, pages 1001–1004, 2004.

[33] International Business Machines Corporation
`http://www.ibm.com/`.

[34] S. Ishikawa, T. Ikeda, K. Miki, F. Adachi, R. Isotani, K. Iso, and A. Okumura. Speech-activated Text Retrieval System for Multimodal Cellular Phones. In *Proc. of ICASSP*, pages 453–456, 2004.

[35] K. Ito, M. Yamamoto, K. Takeda, T. Takezawa, T. Matsuoka, T. Kobayashi, K. Shikano, and S. Itahashi. JNAS: Japanese Speech Corpus for Large Vocabulary Continuous Speech Recognition Research. *The Journal of the Acoustical Society of Japan*, 20:199–206, 1999.

[36] F. Jelinek. *Statistical Methods for Speech Recognition.* Massachusetts Institute of Technology, 1997.

[37] Julius, an Open-Source Large Vocabulary CSR Engine `http://julius.sourceforge.jp/`.

[38] T. M. Kamm and G. G. L. Meyer. Robustness Aspects of Active Learning for Acoustic Modeling. In *Proceedings of the International Conference on Spoken Language Processing*, pages 1095–1098, 2004.

[39] H. Kawanami, M. Kida, N. Hayakawa, T. Cincarek, T. Kitamura, T. Kato, and K. Shikano. Spoken Guidance Systems Kita-chan and Kita-chan Robot. Their Development and Operation in a Railway Station. Technical report, IEICE, SP2006-14, 2006.

[40] T. Kemp and A. Waibel. Unsupervised Training of a Speech Recognizer: Recent Experiments. In *European Conference on Speech Communication and Technology*, pages 2725–2728, 1999.

[41] L. Lamel, J.-L. Gauvain, and G. Adda. Lightly Supervised Acoustic Model Training. In *ISCA ITRW Adaptation Methods for Speech Recognition*, pages 150–154, 2000.

[42] L. Lamel, J.-L. Gauvain, and G. Adda. Unsupervised Acoustic Model Training. In *Proc. of ICASSP*, volume 1, pages 877–880, 2002.

[43] L. Lamel, F. Lefevre, J.-L. Gauvain, and G. Adda. Portability Issues for Speech Recognition Technologies. In *Proc. of HLT*, pages 9–16, 2001.

[44] L. Lamel, S. Rosset, J.L. Gauvain, S. Bennacef, M. Garnier-Rizet, and B. Routs. The LIMSI ARISE system. *Speech Communication*, 4(31):339–353, 2000.

[45] A. Lee, T. Kawahara, K. Takeda, and K. Shikano. A New Phonetic Tied-Mixture Model for Efficient Decoding. In *International Conference on Acoustics, Speech, and Signal Processing*, pages 1269–1272, 2000.

[46] A. Lee, K. Nakamura, R. Nisimura, H. Saruwatari, and K. Shikano. Noise-Robust Real-World Spoken Dialogue System using GMM-Based Rejection of Unintended Inputs. In *Proc. of ICSLP*, pages pp. 173–176, 2004.

[47] F. Lefevre, J.-L. Gauvain, and L. Lamel. Genericity and Portability for Task-independent Speech Recognition. *Computer Speech and Language*, 19:345–363, 2005.

[48] C. Leggetter and P. Woodland. Maximum Likelihood Linear Regression for Speaker Adaptation of Continuous Density Hidden Markov Models. *Computer Speech and Language*, 9:171–185, 1995.

[49] J. Ma, S. Matsoukas, O. Kimball, and R. Schwartz. Unsupervised training on large amounts of broadcast news data. In *Proc. of ICASSP*, volume 3, pages 1056–1059, 2006.

[50] L. Mangu, E. Brill, and A. Stolke. Finding Consensus Among Words: Lattice-Based Word Error Minimization. In *Proc. of Eurospeech*, pages 495–498, 1999.

[51] T. K. Moon. The Expectation-Maximization Algorithm. *IEEE Signal Processing Magazine*, pages 47–60, Nov. 2006.

[52] G. Nagino and M. Shozakai. Building and Effective Corpus by Using Acoustic Space Visualization (COSMOS) Method. In *Proc. of ICASSP*, volume 1, pages 449–452, 2005.

[53] R. Nishimura, Y. Nishihara, R. Tsurumi, A. Lee, H. Saruwatari, and K. Shikano. Takemaru-kun: Speech-oriented Information System for Real World Research Platform. In *International Workshop on Language Understanding and Agents for Real World Interaction*, pages 70–78, 2003.

[54] R. Nisimura, A. Lee, H. Saruwatari, and K. Shikano. Public Speech-oriented Guidance System with Adult and Child Discrimination Capability. In *Proc. of ICASSP*, pages 433–436, 2004.

[55] R. Nisimura, A. Lee, M. Yamada, and K. Shikano. Operating a Public Spoken Guidance System in Real Environment. In *European Conference on Speech Communication and Technology*, pages 845–848, 2005.

[56] Y. Normandin. MMIE Training for Large Vocabulary Continuous Speech Recognition. In *Proc. of ICASSP*, pages 1367–1371, 1994.

[57] Nuance, Supplier of ASR, Imaging, PDF and OCR Solutions
`http://www.nuance.com/`.

[58] Publicly Available Language Modeling Toolkit
`http://palmkit.sourceforge.net/`.

[59] Joseph W. Picone. Signal modeling techniques in speech recognition. *Proceedings of the IEEE*, 81(9):1215–1247, 1993.

[60] A. Potamianos, S. Narayanan, and S. Lee. Automatic Speech Recognition for Children. In *Proc. of Eurospeech*, pages 2371–2374, 1997.

[61] L. Rabiner and B.-H. Juang. *Fundamentals of Speech Recognition.* Prentice Hall, Englewood Cliffs, NJ, 1993.

[62] L. R. Rabiner. A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition. In *Proceedings of the IEEE*, volume 77, pages 257–286, 1989.

[63] A. Raux, D. Bohus, B. Langner, A.W. Black, and M. Eskenazi. Doing Research on a Deployed Spoken Dialogue System: One Year of Let's Go! Experience. In *Proceedings of the International Conference on Spoken Language Processing*, pages 65–68, 2006.

[64] D.A. Reynolds. Speaker Identification and Verification using Gaussian Mixture Speaker Models. *Speech Communication*, 17:91–108, 1995.

[65] G. Riccardi and D. Hakkani-Tür. Active and Unsupervised Learning for Automatic Speech Recognition. In *European Conference on Speech Communication and Technology*, pages 1825–1828, 2003.

[66] R. Sarikaya, A. Gravano, and Y. Gao. Rapid Language Model Development using External Resources for New Spoken Dialogue Domains. In *Proc. of ICASSP*, pages 573–576, 2005.

[67] T. Schultz and A. Waibel. Multilingual and Crosslingual Speech Recognition. In *Proc. of DARPA Workshop on Broadcast News Transcription and Understanding*, pages 259–262, 1998.

[68] S. Seneff and J. Polifroni. Dialogue Management in the Mercury Flight Reservation System. In *Proceedings of NASLP-NAACL Satellite Workshop*, pages 1–6, 2000.

[69] A. Sethy, S. Narayanan, and B. Ramabhadran. Data-Driven Approach for Language Model Adaptation using Stepwise Relative Entropy Minimization. In *Proc. of ICASSP*, volume 4, pages 177–180, 2007.

[70] T.T. Soong. *Fundamentals of Probability and Statistics for Engineers.* John Wiley & Sons, Ltd., 2004.

[71] SpinVox, Speech-to-Text Conversion Services
`http://www.spinvox.com/`.

[72] A. Stolcke. SRILM - An Extensible Language Modeling Toolkit. In *Proc. of ICSLP*, pages 901–904, 2002.

[73] E. Sumita, T. Shimizu, and S. Nakamura. NICT-ATR Speech-To-Speech Translation System. In *Proc. of the ACL*, pages 25–28, 2007.

[74] S. Takeuchi, T. Cincarek, H. Kawanami, H. Saruwatari, and K. Shikano. Construction and optimization of a question and answer database for a real-environment speech-oriented guidance system. In *Proc. of Oriental COCOSDA*, 2007.

[75] M. Tang, B. Pellom, and K. Hacioglu. Call-Type Classification and Unsupervised Training for the Call Center Domain. In *IEEE Workshop on Automatic Speech Recognition and Understanding*, pages 204–208, 2003.

[76] G. Tur. Multitask Learning for Spoken Language Understanding. In *Proc. of ICASSP*, pages 585–588, 2006.

[77] G. Tur, R.E. Shapire, and D. Hakkani-Tür. Active Learning for Spoken Language Understanding. In *International Conference on Acoustics, Speech, and Signal Processing*, volume 1, pages 276–279, 2003.

[78] IBM, ViaVoice
`http://www.nuance.com/viavoice/`.

[79] K. Visweswariah, R. Gopinath, and V. Goel. Task Adaptation of Acoustic and Language Models based on Large Quantities of Data. In *Proc. of Interspeech*, pages 1284–1287, 2004.

[80] F. Wessel and H. Ney. Unsupervised Training of Acoustic Models for Large Vocabulary Continuous Speech Recognition. In *IEEE Workshop on Automatic Speech Recognition and Understanding*, 2001.

[81] S. Witt and S. Young. Off-line acoustic modelling of non-native accents. In *Proc. of Eurospeech*, volume 3, pages 1367–1370, 1999.

[82] S. Yoshizawa, A. Baba, K. Matsunami, Y. Mera, M. Yamada, A. Lee, and K. Shikano. Evaluation on Unsupervised Speaker Adaptation based on Sufficient HMM Statistics of Selected Speakers. In *European Conference on Speech Communication and Technology*, pages 1219–1222, 2001.

[83] R. Zhang, Z.A. Bawab, A. Chan, D. Huggins-Daines A. Chotimongkol, and A.I. Rudnicky. Investigations on ensemble-based semi-supervised acoustic model training. In *Proc. of Interspeech*, pages 1677–1680, 2005.

[84] R. Zhang and A. I. Rudnicky. A New Data Selection Approach for Semi-Supervised Acoustic Modeling. In *Proc. of ICASSP*, volume 1, pages 421–424, 2006.

[85] V. Zue, J. Class, D. Goodine, H. Leung, M. Phillips, J. Polifroni, and S. Seneff. Preliminary ATIS Development at MIT. In *Proc. of Speech and Natural Language Workshop held at Hidden Valley, Pennsylvania*, pages 130–135. Morgan Kaufmann Publishers, Inc., 1990.

Publications

Journal Papers

1. Tobias Cincarek, Tomoki Toda, Hiroshi Saruwatari, Kiyohiro Shikano, "Utterance-based Selective Training for the Automatic Creation of Task-Dependent Acoustic Models", IEICE Trans.Information and Systems, Vol.E89-D, No.3, pp.962-969, 2006.

2. Tobias Cincarek, Tomoki Toda, Hiroshi Saruwatari, Kiyohiro Shikano, "Cost Reduction of Acoustic Modeling for Real-Environment Applications Using Unsupervised and Selective Training", IEICE Trans. Information and Systems, Special Section on Robust Speech Processing in Realistic Environments, March 2008.

3. Tobias Cincarek, Hiromichi Kawanami, Ryuichi Nisimura, Akinobu Lee, Hiroshi Saruwatari, Kiyohiro Shikano, "Development, Long-Term Operation and Portability of a Real-Environment Speech-oriented Guidance System", IEICE Trans. Information and Systems, Special Section on Robust Speech Processing in Realistic Environments, March 2008.

International Conferences

1. Tobias Cincarek, Tomoki Toda, Hiroshi Saruwatari, Kiyohiro Shikano, "Selective EM Training of Acoustic Models based on Sufficient Statistics of Single Utterances", IEEE Automatic Speech Recognition and Understanding Workshop (ASRU), pp. 168-173, November 2005.

2. Tobias Cincarek, Tomoki Toda, Hiroshi Saruwatari, Kiyohiro Shikano, "Utterance-based Selective Training for Cost-effective Task-adaptation of Acoustic Models," Workshop on Speech Recognition and Intrinsic Variation (SRIV), pp. 71-76, May 2006.

3. Tobias Cincarek, Tomoki Toda, Hiroshi Saruwatari, Kiyohiro Shikano, "Acoustic Modeling for Spoken Dialogue Systems Based on Unsupervised Utterance-based Selective Training," the 9th International Conference on

Spoken Language Processing (Interspeech 2006 - ICSLP), pp. 1722-1725, Sept. 2006.

4. Tobias Cincarek, Ryuichi Nisimura, Akinobu Lee, Kiyohiro Shikano, "Insights Gained from Development and Long-Term Operation of a Real-Environment Speech-Oriented Guidance System", International Conference on Acoustics, Speech, and Signal Processing (ICASSP), Vol. 4, pp. 157-160, Apr. 2007.

5. Tobias Cincarek, Izumi Shindo, Tomoki Toda, Hiroshi Saruwatari, Kiyohiro Shikano, "Development of Preschool Children Subsystem for ASR and QA in a Real-Environment Speech-oriented Guidance Task", Proceedings of the 10th European Conference on Speech Communication and Technology (Interspeech 2007 - Eurospeech), pp. 1469-1472, Aug. 2007.

6. Tobias Cincarek, Hiromichi Kawanami, Hiroshi Saruwatari, Kiyohiro Shikano, "Development and Portability of ASR and Q&A Module for Real-Environment Speech-Oriented Guidance Systems", IEEE Automatic Speech Recognition and Understanding Workshop (ASRU), pp. 520-525, Dec. 2007.

7. Christian Hacker, Anton Batliner, Stefan Steidl, Elmar Nöth, Heinrich Niemann, Tobias Cincarek, "Assessment of Non-Native Children's Pronunciation: Human Marking and Automatic Scoring", 10th International Conference on Speech and Computer, Vol. 1, pp. 123-126, 2005.

8. Christian Hacker, Tobias Cincarek, Andreas Maier, Andre Heßler, Elmar Nöth, "Boosting of Prosodic and Pronunciation Features to Detect Mispronunciations of Non-Native Children", International Conference on Acoustics, Speech, and Signal Processing (ICASSP), Vol. 4, pp. 197-200, Apr. 2007.

9. Hiroyuki Sakai, Hiromichi Kawanami, Tobias Cincarek, Kiyohiro Shikano, Hiroshi Saruwatari, Akinobu Lee, "Voice Activity Detection Applied to Hands-Free Spoken Dialogue Robot based on Decoding using Acoustic and Language Model", 1st International Conference on Robot Control and Coordination (ROBOCOMM), Oct. 2007.

10. Hiromichi Kawanami, Tobias Cincarek, Takeuchi Shota, Hiroshi Saruwatari, Kiyohiro Shikano, "Development and Operational Result of Real-Environment Speech-Oriented Guidance Systems Kita-robo and Kita-chan", Oriental COCOSDA, Nov. 2007.

11. Shota Takeuchi, Tobias Cincarek, Hiromichi Kawanami, Hiroshi Saruwatari, Kiyohiro Shikano, "Construction and Optimization of a Question and Answer Database for a Real-Environment Speech-Oriented Guidance System", Oriental COCOSDA, Nov. 2007.

Domestic Meetings and Conferences

1. Christian Hacker, Tobias Cincarek, Rainer Gruhn, Stefan Steidl, Elmar Nöth, and Heinrich Niemann, “Pronunciation Feature Extraction”, 27th Annual Meeting of the German Association for Pattern Recognition (DAGM), pp. 141-148, 2005.

2. Tobias Cincarek, Tomoki Toda, Hiroshi Saruwatari, Kiyohiro Shikano, “Selective Training for Acoustic Models based on Sufficient Statistics of Single Utterances”, ASJ Fall Meeting, 1-P-11, pp. 173-174, September 2005 (in Japanese).

3. Tobias Cincarek, Tomoki Toda, Hiroshi Saruwatari, Kiyohiro Shikano, “Utterances-based Maximum Likelihood Selective Training”, 8th Young Researchers Exchange and Presentation Meeting, Kansai Section, Acoustical Society of Japan, December 2005 (in Japanese).

4. Tobias Cincarek, Tomoki Toda, Hiroshi Saruwatari, Kiyohiro Shikano, “Utterance-based Selective Training for Task-Dependent Acoustic Modeling”, IEICE Technical Report, SP2005-135(2005-12), pp. 145-150, December 2005 (in Japanese).

5. Tobias Cincarek, Tomoki Toda, Hiroshi Saruwatari, Kiyohiro Shikano, “Acoustic Model Construction based on Unsupervised Selective Training”, ASJ Spring Meeting, 1-P-13, pp.169-170, March 2006 (in Japanese).

6. Tobias Cincarek, Tomoki Toda, Hiroshi Saruwatari, Kiyohiro Shikano, “Evaluation of Maximum Likelihood Utterance-based Selective Training for Context-dependent Acoustic Models”, ASJ Spring Meeting, 1-P-14, pp.171-172, March 2006 (in Japanese).

7. Tobias Cincarek, Hiromichi Kawanami, Hiroshi Saruwatari, Kiyohiro Shikano, “Preliminary Evaluation of Acoustic Models Built for the Speech-oriented Guidance System Kita-chan”, ASJ Fall Meeting, 1-2-10, pp.19-20, September 2006 (in Japanese).

8. Tobias Cincarek, Hiromichi Kawanami, Manabu Kida, Hiroshi Saruwatari, Kiyohiro Shikano, Ryuichi Nishimura, Akinobu Lee, “Speech-oriented Guidance System Takemaru and its Portability”, IEICE Technical Report, NLC2006-58, SP2006-114, pp. 173-178, December 2006 (in Japanese).

9. Tobias Cincarek, Hiromichi Kawanami, Hiroshi Saruwatari, Kiyohiro Shikano, “Portability of the Speech-oriented Guidance System Takemaru for the Kita Environment”, ASJ Fall Meeting, 2-3-9, pp.77-79, September 2007 (in Japanese).

10. Kiyohiro Shikano, Tobias Cincarek, Tomoyuki Kato, "Speech Dialog System Operation and Acoustic Model Training based on Sufficient Statistics", Workshop on Information-based Induction Sciences (IBIS2005), pp.291-296, October 2005 (Invited Talk) (in Japanese).

11. Izumi Shindo, Tobias Cincarek, Tomoki Toda, Hiroshi Saruwatari, Kiyohiro Shikano, "Utterance-based Selective Training of Acoustic Model for a Public Guidance System from JNAS Database", ASJ Spring Meeting, 1-11-14, pp.49-50, March 2006 (in Japanese).

12. Naoki Hayakawa, Manabu Kida, Tobias Cincarek, Hiromichi Kawanami, Hiroshi Saruwatari, Kiyohiro Shikano, "Improvement of Scoring for Response Selection for a Speech-oriented Guidance System" ASJ Fall Meeting, 3-2-8, pp.87-88, September 2006 (in Japanese).

13. Izumi Shindo, Tobias Cincarek, Tomoki Toda, Hiroshi Saruwatari, Kiyohiro Shikano, "Spontaneous Preschool Children Speech Recognition Considering Pronunciation Variations", ASJ Fall Meeting, 1-2-9, pp.17-18, September 2006 (in Japanese).

14. Hiromichi Kawanami, Manabu Kida, Naoki Hayakawa, Tobias Cincarek, Takahiro Kitamura, Tomoyuki Kato, Kiyohiro Shikano, "Spoken Guidance Systems "Kita-chan" and "Kita-chan Robot", IEICE Technical Report, Vol.106, No.123, SP2006-14, pp. 19-24, June 2006 (in Japanese).

15. Kiyohiro Shikano, Cincarek Tobias, Hiromichi Kawanami, Ryuichi Nishimura, Akinobu Lee, "Development and Evaluation of Takemaru-kun Spoken Guidance System and Portability to Kita-chan and Kita-robo Systems", IPSJ SIG Technical Report, 2006-SLP-63-(7), pp. 33–38, October 2006 (Invited Talk) (in Japanese).

16. Izumi Shindo, Tobias Cincarek, Tomoki Toda, Hiroshi Saruwatari, Kiyohiro Shikano, "Development and Evaluation of Preschool Children Speech Recognition Module of a Public Guidance System", HI122SLP65-19, pp.103-108, February 2007 (in Japanese).

17. Izumi Shindo, Tobias Cincarek, Tomoki Toda, Hiroshi Saruwatari, Kiyohiro Shikano, "Development and Evaluation of Recognition Module of Spontaneous Preschool Children's Speech for a Guidance System", ASJ Spring Meeting, 2-9-1, pp.37-38, March 2007 (in Japanese).

18. Naoki Hayakawa, Tobias Cincarek, Hiromichi Kawanami, Hiroshi Saruwatari, Kiyohiro Shikano, "Introducing Rejection Strategy using Score Threshold for Response Selection of a Speech-oriented Guidance System" ASJ Spring Meeting, 1-P-26, pp.175-176, March 2007 (in Japanese).

19. Shota Takeuchi, Tobias Cincarek, Manabu Kita, Naoki Hayakawa, Hiromichi Kawanami, Hiroshi Saruwatari, Kiyohiro Shikano, "Question Answer Database Optimization based on Leave-one-out Cross-Validation", IEICE Technical Report, SP2007-13, vol.107, no.116, pp. 31-36, June 2007 (in Japanese).

20. Hiroyuki Sakai, Tobias Cincarek, Hiromichi Kawanami, Hiroshi Saruwatari, Kiyohiro Shikano, Akinobu Lee, "Voice Activity Detection Applied to Hands-free Speech Recognition based on Decoding using Acoustic and Language Models", IEICE Technical Report, SP2007-17(2007-06), vol.107, no.116, pp. 55-60, June 2007 (in Japanese).

21. Hiroyuki Sakai, Tobias Cincarek, Hiromichi Kawanami, Hiroshi Saruwatari, Kiyohiro Shikano, Akinobu Lee, "Evaluation of VAD Applied to Hands-free Speech Recognition based on Decoding using Acoustic and Language Models", ASJ Fall Meeting, 3-3-12, pp.167-168, September 2007 (in Japanese).

22. Shota Takeuchi, Tobias Cincarek, Hiromichi Kawanami, Hiroshi Saruwatari, Kiyohiro Shikano, "Iteration of Optimization for Question Answer Database based on Leave-one-out Cross-Validation", ASJ Fall Meeting, 3-Q-14, pp.241-242, September 2007 (in Japanese).