STATISTICAL MODELS OF MACHINE TRANSLATION, SPEECH RECOGNITION, AND SPEECH SYNTHESIS FOR SPEECH－TO-SPEECH TRANSLATION

DOCTORAL DISSERTATION

STATISTICAL MODELS OF MACHINE TRANSLATION, SPEECH RECOGNITION, AND SPEECH SYNTHESIS FOR SPEECH-TO-SPEECH TRANSLATION

DOCTOR OF ENGINEERING

JANUARY 2011

Kei HASHIMOTO

Supervisor : Dr. Keiichi TOKUDA

Department of Scientific and Engineering Simulation
Nagoya Institute of Technology

Abstract

In speech-to-speech translation, the source language speech is translated into target language speech. A speech-to-speech translation system can help to overcome the language barrier, and is essential for providing more natural interaction. A speech-to-speech translation system consists of three components: speech recognition, machine translation and speech synthesis. In order to improve the end-to-end performance of the speech-to-speech translation system, it is required to improve the performance of each component. Recently, statistical approaches are widely used in these fields. In this paper, statistical models for improving the performance of speech-to-speech translation systems are proposed.

First, a reordering model using a source-side parse-tree for phrase-based statistical machine translation is proposed. In the proposed method, the target-side word order is obtained by rotating nodes of the source-side parse-tree. The node rotation (monotone or swap) is modeled using word alignments based on a training parallel corpus and source-side parse-trees. The model efficiently suppresses erroneous target word orderings, especially global orderings. In English-to-Japanese and English-to-Chinese translation experiments, the proposed method resulted in a 0.49-point improvement (29.31 to 29.80) and a 0.33-point improvement (18.60 to 18.93) in word BLEU-4 compared with IST-ITG constraints, respectively. This indicates the validity of the proposed reordering model.

Next, Bayesian context clustering using cross validation for hidden Markov model (HMM) based speech recognition is proposed. The Bayesian approach can select an appropriate model structure while taking account of the amount of training data and can use prior information as prior distributions. Since prior distributions affect estimation of the posterior distributions and selection of model structure, the determination of prior distributions is an important problem. The proposed method can determine reliable prior distributions without any tuning parameters and select an appropriate model structure while taking account of the amount of training data. Continuous phoneme recognition experiments show that the proposed method achieved a higher performance than the conventional methods.

Next, a new framework of speech synthesis based on the Bayesian approach is proposed.

Since acoustic models greatly affect the quality of synthesized speech in HMM-based speech synthesis, it is required to improve acoustic models for improving the performance of speech synthesis. The Bayesian method is a statistical technique for estimating reliable predictive distributions by treating model parameters as random variables. In the proposed framework, all processes for constructing the system can be derived from one single predictive distribution which represents the basic problem of speech synthesis directly. Experimental results show that the proposed method outperforms the conventional one in a subjective test. And also, a speech synthesis technique integrating training and synthesis processes based on the Bayesian framework is proposed. In the Bayesian speech synthesis, all processes are derived from one single predictive distribution which represents the problem of speech synthesis directly. However, it typically assumes that the posterior distribution of model parameters is independent of synthesis data, and this separates the system into training and synthesis parts. In the proposed method, the approximation is removed and an algorithm that the posterior distributions, model structures and synthesis data are iteratively updated is derived. Experimental results show that the proposed method improves the quality of synthesized speech.

Finally, an analysis of the impacts of machine translation and speech synthesis on speech-to-speech translation systems is provided. Many techniques for integration of speech recognition and machine translation have been proposed. However, speech synthesis has not yet been considered. If the quality of synthesized speech is bad, users will not understand what the system said: the quality of synthesized speech is obviously important for speech-to-speech translation and any integration method intended to improve the end-to-end performance of the system should take account of the speech synthesis component. In order to understand the degree to which each component affects performance, a subjective evaluation to analyze the impact of machine translation and speech synthesis components is reported. The results of these analyses show that the naturalness and intelligibility of synthesized speech are strongly affected by the fluency of the translated sentences.

For speech-to-speech translation systems, above techniques were proposed. Experimental results show that the proposed techniques improves the performances and the naturalness and intelligibility of synthesized speech are strongly affected by the fluency of the translated sentences.

Keywords: Speech-to-speech translation, machine translation, reordering model, speech recognition, speech synthesis, Baysian approach,

Abstract in Japanese

近年，社会の国際化に伴い，音声翻訳システムの開発への期待が高まっている．音声翻訳システムとは，音声を入出力とした翻訳システム，つまり，ある言語で発話された音声を他の言語の音声に直接翻訳して出力するシステムである．音声は我々人間にとって最も身近な情報伝達手段であるため，入出力にテキストを用いている従来の翻訳システムと比較して，自然なコミュニケーションが可能であり，このような音声翻訳技術が実現されることより，言語の壁を越えた円滑なコミュニケーションが可能となる．音声翻訳システムは音声認識部，機械翻訳部，音声合成部の 3 つの要素から構成される．近年，機械翻訳，音声認識，音声合成の各分野において，統計モデルに基づく手法が注目を集めいる．統計モデルに基づく機械翻訳，音声認識，音声合成は，あらゆる言語のシステムを同様の枠組みを用いてシステムを構築することができるため，多言語への対応が容易であり，音声翻訳システムに適しているといえる．これまでは，各要素が独立した形でシステムが構成されていたが，今後は音声翻訳の問題を一つの大きな統計問題として捉え，音声翻訳全体を考慮した形で統計モデルの最適化を行うべきと考えている．しかし，各要素の性能はまだまだ十分なものとは言えず，音声翻訳システムの実現のためには各要素のさらなる高性能化が必要不可欠である．本論文では音声翻訳システムのための，より高性能な統計モデルの提案を目的とする．

まず，統計的機械翻訳のための入力文の構文木を用いた単語並び替えモデルを提案する．統計的機械翻訳の分野において，翻訳結果の大局的な単語並び替えの問題は最も重要な問題の一つである．この問題に対し，近年，構文情報を用いた統計的機械翻訳手法に注目を集めている．提案法では，入力文の構文木を回転させることによって翻訳文の構文木を表現することが可能であると仮定し，構文木の回転を，学習データの単語アライメントと入力文の品詞を用いることによってモデル化する．提案モデルは，構文情報を考慮しているため，特に大局的な語順に効果を発揮するモデルであるといえる．英日翻訳実験において，自動評価尺度 BLEU-4 が先行研究から 0.49 改善され，提案法の有効性が確認された．

次に，ベイズ基準による隠れマルコフモデル (Hidden Markov Model; HMM) に基づく音声認識におけるクロスバリデーションを用いたモデル構造選択手法を提案する．

ベイズ基準では，学習データ量を考慮した適切なモデル構造を選択することが可能であるという利点がある．しかし，ベイズ基準では，事前分布は任意に設定することが可能であり，モデル構造選択において事前分布パラメータは調整パラメータのように働くため，適切な事前分布を設定することは，ベイズ基準によるモデル構造選択において重要な問題である．本論文では，クロスバリデーションを用いた事前分布設定方法を提案し，ベイズ基準によるモデル構造選択に適用した．連続音素認識実験において，提案法は従来法から音素認識率を改善し，提案法が適切なモデルを選択することが可能であることを示した．

次に，ベイズ基準による HMM 音声合成手法を提案する．HMM 音声合成によって出力される合成音声は音響モデルに強く影響を受けるため，高品質な合成音声を生成するためには，高精度な音響モデルを推定することが必要不可欠である．提案法は，学習データから合成データが出力されるという，音声合成の問題を直接表す予測分布から音声合成システム全体を表現する，全く新しい音声合成手法である．主観評価実験から，ベイズ基準による HMM 音声合成手法は従来法から合成音声の品質を改善することを示した．さらに，学習・合成過程が統合されたベイズ基準による HMM 音声合成を提案する．これまで，ベイズ基準による HMM 音声合成では，合成データは事後分布に対し独立であるという近似を用いており，学習・合成過程が分離されていた．しかし，このような分離は，音声合成の問題を直接表す予測分布から音声合成システム全体を表現するという，ベイズ基準による音声合成の特徴を十分に表現することができていない．提案法では，合成データを用いて事後分布を再推定することにより，近似が排除されたベイズ基準による音声合成を実現する．主観評価実験により，提案法は合成音声の品質を改善することを示した．

最後に，音声翻訳システムにおける機械翻訳部，音声合成部の影響について調査，分析する．これまでに，音声翻訳システムのための音声認識部，機械翻訳部の統合手法が数多く提案されてきた．しかし，これらの手法では音声合成部は考慮されていなかった．合成音声の品質が十分な品質でなかった場合，システムの利用者は合成音声の発話内容を理解することができないため，音声音訳システムにおいて，音声合成部は非常に重要な要素である．このため，音声合成部を考慮した統合手法が今後必要となると考えられる．本論文では，機械翻訳部，音声合成部に注目した主観評価実験を行い，音声翻訳システムにおけるそれぞれの要素の影響について調査，分析した．分析結果から，機械翻訳部の出力する翻訳文が流暢な文であるほど，合成音声の品質は改善され，また被験者の単語聞き取り精度が改善されることが確認された．

以上のように，本論文では，音声翻訳システムにおける，音声認識，機械翻訳，音声合成のためのより高性能なモデル化手法を提案し，これらの手法の有効性を示す．また，機械翻訳と音声合成の影響の分析を行い，統合手法の検討を行った．

Acknowledgement

First of all, I would like to express my sincere gratitude to Keiichi Tokuda, my advisor, for his support, encouragement, and guidance.

I would like to thank Akinobu Lee, Yoshihiko Nankaku, and Heiga Zen (currently with Toshiba Research Europe) for their technical supports and helpful discussions. Special thanks go to all the members of Tokuda and Lee laboratories for their technical support and encouragement. If somebody was missed among them, my work would not be completed. I would be remiss if I did not thank Natsuki Kuromiya, a secretary of the laboratory, for their kind assistance.

I am grateful to Satoshi Nakamura (with NICT), Eiichiro Sumita (with NICT), Hirofumi Yamamoto (with Kinki University), and Hideo Okuma (with NICT), for giving me the opportunity to work in ATR Spoken Language Communication Research Laboratories, and for their valuable advice.

I am also grateful to Simon King (with University of Edinburgh), William Byrne (with University of Cambridge), and Junichi Yamagishi (with University of Edinburgh), for giving me the opportunity to work in University of Edinburgh and University of Cambridge, and for their valuable advice.

Finally, I would sincerely like to thank my parents and my friends for their encouragement.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

In speech-to-speech translation (S2ST), the source language speech is translated into target language speech. A S2ST system can help to overcome the language barrier, and is essential for providing more natural interaction. A S2ST system consists of three components: speech recognition, machine translation and speech synthesis. Figure 1.1 shows the overview of a S2ST system. In order to improve the end-to-end performance of the S2ST system, it is required to improve the performance of each component. Recently, statistical approaches are widely used in these fields. In this paper, statistical models for improving the performance of S2ST systems are proposed.

Statistical machine translation has been widely applied in many state-of-the-art translation systems. A popular statistical machine translation paradigms is the phrase-based statistical machine translation [1, 2]. In phrase-based statistical machine translation, errors in word reordering, especially global reordering, are one of the most serious problems. To resolve this problem, many word-reordering constraint techniques have been proposed. In inversion transduction grammar (ITG) constraints [3, 4], the target-side word order is obtained by rotating nodes of the source-side binary tree. In these node rotations, the source binary tree instance is not considered. Imposing a source tree on ITG (IST-ITG) constraints [5] is an extension of ITG constraints and a hybrid of the first and second type of approach. IST-ITG constraints directly introduce a source sentence tree structure. Therefore, IST-ITG can obtain stronger constraints for word reordering than the original ITG constraints. Although IST-ITG constraints efficiently suppress erroneous target word orderings, the method cannot assign the probability to the target word orderings. In this paper, a reordering model using a source-side parse-tree for phrase-based statistical machine translation is proposed. The proposed reordering model is an extension of IST-ITG constraints. In the proposed method, the target-side word order is obtained by rotating nodes of a source-side parse-tree in a similar fashion to IST-ITG constraints. The rotating

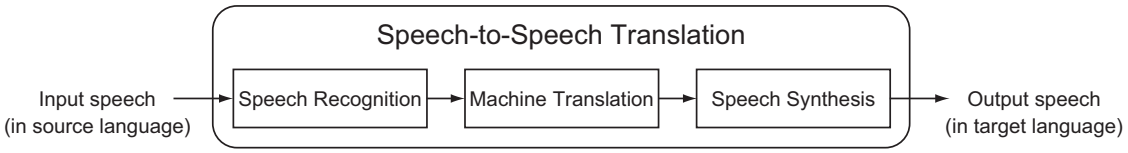

Figure 1.1: Overview of a speech-to-speech translation system

positions, monotone or swap, are modeled from word alignments of a training parallel corpus and source-side parse-trees. The proposed method can conduct a probabilistic evaluation of target word orderings using the source-side parse-tree.

In the field of speech recognition, hidden Markov models (HMMs) have been widely used as acoustic models. In HMM-based speech recognition systems [6], accurate acoustic modeling is necessary for reducing recognition error rate. The maximum likelihood (ML) criterion is one of the standard criteria for training acoustic models in speech recognition. The ML criterion guarantees to estimate the true values of the parameters as the amount of training data infinitely increases. However, since the ML criterion produces a point estimate of model parameters, the estimation accuracy may be degraded due to the over-fitting problem when the amount of training data is insufficient. On the other hand, the Bayesian approach considers the posterior distribution of all variables [7]. That is, all the variables introduced when models are parameterized, such as model parameters and latent variables, are regarded as random variables, and their posterior distributions are obtained based on the Bayes theorem. Based on this posterior distribution estimation, the Bayesian approach can generally achieve more robust model construction and classification than the ML approach [8–10]. And also, the Bayesian approach can select an appropriate model structure [11, 12], even when there are insufficient amounts of data. Therefore, the speech recognition framework based on the Bayesian approach is effective for estimating appropriate acoustic models and model structures. Moreover, the Bayesian approach can utilize prior distributions which represent the prior information of model parameters. In the Bayesian approach, since prior distributions of model parameters affect the estimation of posterior distributions and model selection, the determination of prior distributions is an important problem for estimating appropriate acoustic models. In this paper, a prior distribution determination technique using cross validation is proposed and it is applied to the context clustering for the speech recognition framework based on Bayesian approach. The cross validation method is known as a straightforward and useful method for model structure optimization [13, 14]. The main idea behind cross validation is to split data for estimating the risk of each model. Part of data is used for training each model, and the remaining part is used for estimating the risk of the model. Then, the cross validation method selects the model with the smallest estimated risk. The cross validation method avoids the over-fitting problem because the training data is independent from the

validation data. The context clustering based on the ML criterion using cross validation has been proposed, and it can select a more appropriate model structure than the conventional ML criterion [15]. The proposed method can be regarded as an extension of context clustering using cross validation to the Bayesian approach. Using prior distributions determined by the cross validation, it is expected that a higher generalization ability is achieved and an appropriate model structure can be selected in the context clustering without any tuning parameters.

A statistical speech synthesis system based on HMMs was recently developed. In HMM-based speech synthesis, the spectrum, excitation and duration of speech are modeled simultaneously with HMMs, and speech parameter sequences are generated from the HMMs themselves [16]. In HMM-based speech synthesis, the ML criterion has been typically used for training HMMs and generating speech parameters. The ML criterion guarantee that the ML estimates approach the true values of the parameters. However, since the ML criterion produces a point estimate of the HMM parameters, its estimation accuracy may deteriorate when the amount of training data is insufficient. To overcome this problem, a Bayesian speech synthesis framework is proposed in this paper. In this framework, all processes for constructing the system are derived from one single predictive distribution which exactly represents the problem of speech synthesis. The Bayesian approach considers the posterior distribution of any variable [7]. That is, all the variables introduced when the models are parameterized, such as the model parameters and latent variables, are regarded as probabilistic variables, and their posterior distributions are obtained by invoking Bayes theorem. Based on the posterior distribution estimation, the Bayesian approach can generally construct a more robust model than the ML approach. However, the Bayesian approach requires complex integral and expectation computations to obtain posterior distributions when the models have latent variables. To overcome this problem, a variational Bayes (VB) method [17] has recently been proposed in the learning theory field. This method can obtain approximate posterior distributions through iterative calculations similar to the expectation-maximization (EM) algorithm used in the ML approach. The proposed method can estimate reliable predictive distributions by marginalizing model parameters.

Furthermore, a Bayesian speech synthesis framework integrating training and synthesis processes is also proposed. In the Bayesian speech synthesis, the estimation of the posterior distributions, model selection, and speech parameter generation are consistently performed by maximizing the log marginal likelihood. The posterior distributions of all variables are obtained by using the VB method. Then, the obtained posterior distribution of the model parameters depends on not only the training data, but also the synthesis data. In a basic speech synthesis situation, the observed data for the synthesis sentences is not given beforehand. Therefore, the posterior distributions cannot be obtained. To overcome

this problem, it typically assumes that the posterior distribution of the model parameters is independent of the synthesis data [18, 19]. As a result of this approximation, the Bayesian speech synthesis system is separated into training and synthesis parts, as the conventional ML-based system, and the posterior distribution of the model parameters and decision trees can be obtained from only the training data. However, although the posterior distributions can be estimated, they don't consider synthesis data, and the system doesn't represent the Bayesian speech synthesis exactly. This paper proposes a speech synthesis technique integrating training and synthesis processes based on the Bayesian framework. This method removes the approximation and leads to an algorithm that the posterior distributions, decision trees and synthesis data are iteratively updated.

Finally, an analysis of the impacts of machine translation and speech synthesis on speech-to-speech translation systems is provided. In the simplest S2ST system, only the single-best output of one component is used as input to the next component. Therefore, errors of the previous component strongly affect the performance of the next component. Due to errors in speech recognition, the machine translation component cannot achieve the same level of translation performance as achieved for correct text input. To overcome this problem, many techniques for integration of speech recognition and machine translation have been proposed, such as [20, 21]. However, the speech synthesis component is not usually considered. The output speech for translated sentences is generated by the speech synthesis component. If the quality of synthesized speech is bad, users will not understand what the system said: the quality of synthesized speech is obviously important for S2ST and any integration method intended to improve the end-to-end performance of the system should take account of the speech synthesis component. This paper focuses on the impact of the machine translation and speech synthesis components on end-to-end performance of an S2ST system. In order to understand the degree to which each component affects performance, we investigate integration methods. First, a subjective evaluation divided into three sections: speech synthesis, machine translation, and speech-to-speech translation, is conducted. Various translated sentences were evaluated by using N-best translated sentences output from the machine translation component. The individual impacts of the machine translation and the speech synthesis components are analyzed from the results of this subjective evaluation.

For speech-to-speech translation, above improved techniques were proposed and systems using these techniques improved their performance. The rest of the present dissertation is organized as follows. Chapter 2 introduces reordering model using source-side parse-tree for statistical machine translation. Chapter 3 shows Bayesian context clustering using cross validation for speech recognition. Chapter 4 presents Bayesian speech synthesis and integration technique of training and synthesis processes for Bayesian speech synthesis. An analysis of the impacts of machine translation and speech synthesis on speech-to-

speech translation systems is provided in Chapter 5. Concluding remarks and future plans are presented in the final chapter.

Chapter 2

A Reordering Model Using a Source-Side Parse-Tree for Statistical Machine Translation

Statistical machine translation has been widely applied in many state-of-the-art translation systems. A popular statistical machine translation paradigms is the phrase-based statistical machine translation [1, 2]. In phrase-based statistical machine translation, errors in word reordering, especially global reordering, are one of the most serious problems. To resolve this problem, many word-reordering constraint techniques have been proposed. These techniques are categorized into two types. The first type is linguistically syntax-based. In this approach, tree structures for the source [22, 23], target [24, 25], or both [26] are used for model training. The second type is formal constraints on word permutations. IBM constraints [27], the lexical word reordering model [28], and inversion transduction grammar (ITG) constraints [3, 4] belong to this type of approach. For ITG constraints, the target-side word order is obtained by rotating nodes of the source-side binary tree. In these node rotations, the source binary tree instance is not considered. Imposing a source tree on ITG (IST-ITG) constraints [5] is an extension of ITG constraints and a hybrid of the first and second type of approach. IST-ITG constraints directly introduce a source sentence tree structure. Therefore, IST-ITG can obtain stronger constraints for word reordering than the original ITG constraints. For example, IST-ITG constraints allows only eight word orderings for a four-word sentence, even though twenty-two word orderings are possible with respect to the original ITG constraints. Although IST-ITG constraints efficiently suppress erroneous target word orderings, the method cannot assign the probability to the target word orderings.

This chapter presents a reordering model using a source-side parse-tree for phrase-based

statistical machine translation. The proposed reordering model is an extension of IST-ITG constraints. In the proposed method, the target-side word order is obtained by rotating nodes of a source-side parse-tree in a similar fashion to IST-ITG constraints. We modeled the rotating positions, monotone or swap, from word alignments of a training parallel corpus and source-side parse-trees. The proposed method conducts a probabilistic evaluation of target word orderings using the source-side parse-tree.

The rest of this chapter is organized as follows. Section 2.1 describes the previous approach to resolving erroneous word reordering. In Section 2.2, the reordering model using a source-side parse-tree is presented. Section 2.3 shows experimental results. Finally, Section 2.4 presents the summary and some concluding remarks and future works.

2.1 Previous Work

First, we introduce two previous studies on related word reordering constraints, ITG and IST-ITG constraints.

2.1.1 ITG Constraints

In one-to-one word-alignment, the source word f_i is translated into the target word e_i. The source sentence $[f_1, f_2, \cdots, f_N]$ is translated into the target sentence which is the reordered target word sequence $[e_1, e_2, \cdots, e_N]$. Then, the number of reorderings is $N!$.

Stochastic synchronous grammars provide a generative process to produce a sentence and its translation simultaneously. An inversion transduction grammar (ITG) [3, 4] is a well-studied synchronous grammar formalism. To allow for movement during translation, non-terminal productions can be either straight (monotone) or inverted. Straight productions are output in the given order in both sentences. Inverted productions are output in the reverse order in the foreign sentence only. ITG cannot represent all possible permutations of concepts that many occur during translation, because some permutations will require discontinuous constituents. When these ITG constraints are introduced, the number of reorderings $N!$ can be reduced in accordance with the following constraints.

- All possible source-side binary tree structures are generated from the source word sequence.
- The target sentence is obtained by rotating any node of the generated source-side binary trees.

When $N = 4$, the ITG constraints can reduce the number of reorderings from $4! = 24$ to 22 by rejecting the orders $[e_3, e_1, e_4, e_2]$ and $[e_2, e_4, e_1, e_3]$ that cannot be represented by ITG. Such target word orders are called inside-out alignments [4]. For a four-word sentence, the search space is reduced to 92% (22/24), but for a 10-word sentence, the search space is only 6% (206,098/3,628,800) of the original full space.

2.1.2 IST-ITG Constraints

In ITG constraints, the source-side binary tree instance is not considered. Therefore, if a source sentence tree structure is utilized, stronger constraints than the original ITG constraints can be created. IST-ITG constraints [5] directly introduce a source sentence tree structure. The target sentence is obtained with the following constraints.

- A source sentence tree structure is generated from the source sentence.
- The target sentence is obtained by rotating any node of the source sentence tree structure.

By parsing the source sentence, the source-side parse-tree is obtained. After parsing the source sentence, a bracketed sentence is obtained by removing the node syntactic labels; this bracketed sentence can then be converted into a tree structure. For example, the source-side parse-tree "(S1 (S (NP (DT This)) (VP (AUX is) (NP (DT a) (NN pen)))))" is obtained from the source sentence "This is a pen" which consists of four words. By removing the node syntactic labels, the bracketed sentence "((This) ((is) ((a) (pen))))" is obtained. Such a bracketed sentence can be used to produce constraints. If IST-ITG constraints are applied, the number of target word orders in $N = 4$ is reduced to 8, down from 22 with ITG constraints. For example, for the source-side bracketed tree "$((f_1 f_2)\ (f_3 f_4))$," the eight target sequences $[e_1, e_2, e_3, e_4]$, $[e_2, e_1, e_3, e_4]$, $[e_1, e_2, e_4, e_3]$, $[e_2, e_1, e_4, e_3]$, $[e_3, e_4, e_1, e_2]$, $[e_3, e_4, e_2, e_1]$, $[e_4, e_3, e_1, e_2]$, and $[e_4, e_3, e_2, e_1]$ are accepted. For the source-side bracketed tree "$(((f_1 f_2)\ f_3)\ f_4)$," the eight sequences $[e_1, e_2, e_3, e_4]$, $[e_2, e_1, e_3, e_4]$, $[e_3, e_1, e_2, e_4]$, $[e_3, e_2, e_1, e_4]$, $[e_4, e_1, e_2, e_3]$, $[e_4, e_2, e_1, e_3]$, $[e_4, e_3, e_1, e_2]$, and $[e_4, e_3, e_2, e_1]$ are accepted. When the source sentence tree structure is a binary tree, the number of word orderings is reduced to 2^{N-1}. However, the parsing results sometimes do not produce binary trees. In this case, some subtrees have more than two child nodes. For a non-binary subtree, any reordering of child nodes is allowed. If a subtree has three child nodes, six reorderings of the nodes are accepted.

In phrase-based statistical machine translation, a source "phrase" is translated into a target "phrase." However, with IST-ITG constraints, "word" must be used for the constraint unit

since the parse unit is a "word." To absorb different units between translation models and IST-ITG constraints, a new limitation for word reordering is applied.

- Word ordering that destroys a phrase is not allowed.

When this limitation is applied, the translated word ordering is obtained from the bracketed source sentence tree by reordering the nodes in the tree, which is the same as for one-to-one word-alignment.

2.2 Reordering Model Using the Source-Side Parse-Tree

In this section, we present a new reordering model using syntactic information of a source-side parse-tree.

2.2.1 Abstract of Proposed Method

The IST-ITG constraints method efficiently suppresses erroneous target word orderings. However, IST-ITG constraints cannot evaluate the accuracy of the target word orderings; i.e., IST-ITG constraints assign an equal probability to all target word orderings. This chapter proposes a reordering model using the source-side parse-tree as an extension of IST-ITG constraints. The proposed reordering model conducts a probabilistic evaluation of target word orderings using syntactic information of the source-side parse-tree.

In the proposed method, the target-side word order is obtained by rotating nodes of the source-side parse-tree in a similar fashion to IST-ITG constraints. Reordering probabilities are assigned to each subtree of source-side parse-tree S by reordering the positions into two types: monotone (straight) and swap. If the subtree has more than two child nodes, the number of child node order is more than two. However, we assume the child node order other than monotone to be swap.

The source-side parse-tree S consists of subtrees $\{s_1, s_2, \cdots, s_K\}$, where K is the number of subtrees included in the source-side parse-tree. The subtree s_k is represented by the parent node's syntactic label and the order, from sentence head to sentence tail, of the child node's syntactic labels. For example, Figure 2.1 shows a source-side parse-tree for a four-word source sentence consisting of three subtrees. In Figure 2.1, the subtrees s_1, s_2, and s_3 are represented by **S+NP+VP**, **VP+AUX+NP**, and **NP+DT+NN**, respectively. Each subtree has a probability $P(t \mid s)$, where t is monotone (m) or swap (s). The

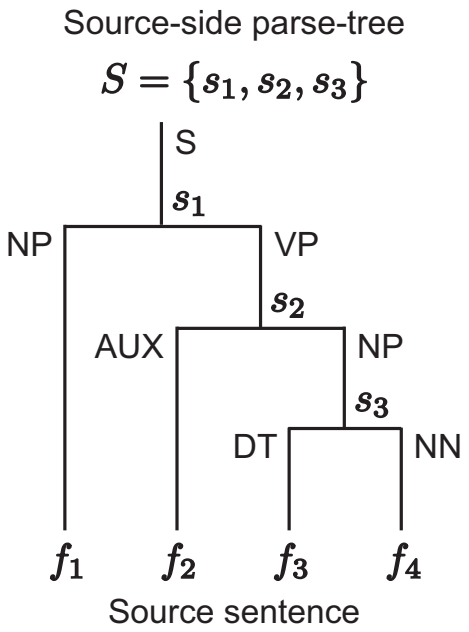

Figure 2.1: Example of a source-side parse-tree of a four-word source sentence consisting of three subtrees.

probability of the target word reordering is calculated as follows.

$$P_r = \prod_{k=1}^{K} P(t \mid s_k) \tag{2.1}$$

By Equation (2.1), each target candidate is assigned the different reordering probability. The proposed reordering probabilities of higher-level subtrees are effective for global word reordering, and ones of lower-level subtrees are effective for local word reordering.

2.2.2 Training of the Proposed Model

We modeled monotone or swap node rotating automatically from word alignments of a training parallel corpus and source-side parse-trees. The training algorithm for the proposed reordering model is as follows.

1. The training process begins with a word-aligned corpus. We obtained the word alignments using Koehn et al.'s method (2003), which is based on Och and Ney's work (2004). This involves running GIZA++ [29] on the corpus in both directions, and applying refinement rules (the variant they designate is "final-and") to obtain a single many-to-many word alignment for each sentence.

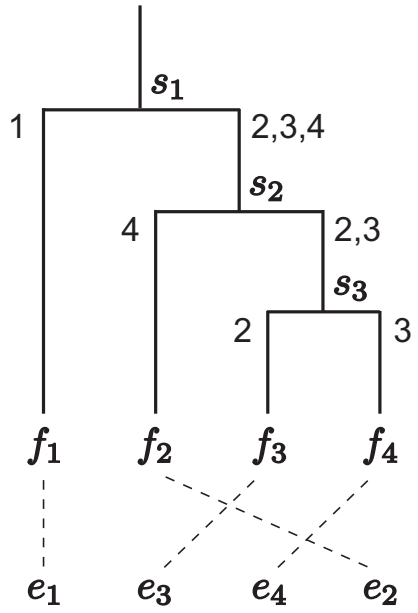

Figure 2.2: Example of a source-side parse-tree with word alignments using the training algorithm of the proposed model.

2. Source-side parse-trees are created using a source language phrase structure parser, which annotates each node with a syntactic label. A source-side parse-tree consists of several subtrees with syntactic labels. For example, the parse-tree "(S1 (S (NP (DT This)) (VP (AUX is) (NP (DT a) (NN pen)))))" is obtained from the source sentence "This is a pen" which consists of four words.

3. Word alignments and source-side parse-trees are combined. Leaf nodes are assigned target word positions obtained from word alignments. Via the bottom-up process, target word positions are assigned to all nodes. For example, in Figure 2.2, the left-side (sentence head) child node of subtree s_2 is assigned the target word position "4," and the right-side (sentence tail) child node is assigned the target word positions "2" and "3," which are assigned to the child nodes of subtree s_3.

4. The monotone and swap reordering positions are checked and counted for each subtree. By comparing the target word positions, which are assigned in the above step, the reordering position is determined. If the target word position of the left-side child node is smaller than one of the right-side child node, the reordering position determined as monotone. For example, in Figure 2.2, the subtrees s_1, s_2 and s_3 are monotone, swap, and monotone, respectively.

5. The reordering probability of the subtree can be directly estimated by counting the

Subtree type	Monotone probability
S+PP+,+NP+VP+.	0.764
PP+IN+NP	0.816
NP+DT+NN+NN	0.664
VP+AUX+VP	0.864
VP+VBN+PP	0.837
NP+NP+PP	0.805
NP+DT+JJ+NN	0.653
NP+DT+JJ+VBP+NN	0.412
NP+DT+NN+CC+VB	0.357

Table 2.1: Example of proposed reordering models.

reordering positions in the training data.

$$P(t \mid s) = \frac{c_t(s)}{\sum_t c_t(s)} \tag{2.2}$$

where $c_t(s)$ is the count of reordering position t included all training samples for the subtree s.

The parsing results sometimes do not produce binary trees. For a non-binary subtree, any reordering of child nodes is allowed. However, the proposed reordering model assumes that reordering positions are only two, monotone and swap. That is, the reordering position which the order of child nodes do not change is monotone, and the other positions are swap. Therefore, the probability of swap $P(\mathrm{s} \mid s_k)$ is derived from the probability of monotone $P(\mathrm{m} \mid s_k)$ as follows.

$$P(\mathrm{s} \mid s_k) = 1.0 - P(\mathrm{m} \mid s_k) \tag{2.3}$$

Table 2.1 shows the example of proposed reordering models.

If a subtree is represented by a binary-tree, there are L^3 possible subtrees, where L is the number of syntactic labels. However, in the possible subtrees, there are subtrees observed only a few times in training sentences, especially when the subtree consists of more than three child nodes. Although a large number of subtree models can capture variations in the training samples, too many models lead to the over-fitting problem. Therefore, subtrees where the number of training samples is less than a heuristic threshold and unseen subtrees are clustered to deal with the data sparseness problem for robust model estimations.

After creating word alignments of a training parallel corpus, there are target word orders which are not derived from rotating nodes of source-side parse-trees. Figure 2.3 shows a

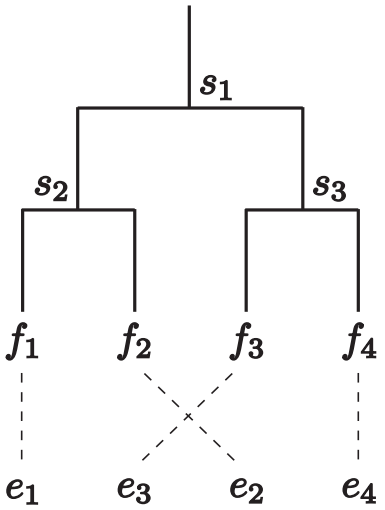

Figure 2.3: Example of a target word order which is not derived from rotating the nodes of source-side parse trees.

sample which is not derived from rotating nodes. Some are due to linguistic reasons, structural differences such as negation (French "ne...pas" and English "not"), adverb, modal and so on. Others are due to non-linguistic reasons, errors of automatic word alignments, syntactic analysis, or human translation [30]. The proposed method discards such problematic cases. In Figure 2.3, the subtree s_1 is then removed from training samples, and the subtrees s_2 and s_3 are used as training samples.

2.2.3 Decoding Using the Proposed Reordering Model

In this section, we describe a one-pass phrase-based decoding algorithm that uses the proposed reordering model in the decoder. The translation target sentence is sequentially generated from left (sentence head) to right (sentence tail), and all reordering is conducted on the source side. To introduce the proposed reordering model into the decoder, the target candidate must be checked for whether the reordering position of a subtree is either monotone or swap whenever a new phrase is selected to extend a target candidate. The checking algorithm is as follows.

1. For old translation candidates, the subtree s, which includes both translated and untranslated words, and its untranslated part u are calculated.

2. When a new target phrase $\bar{e}$ is generated, the source phrase $\bar{f}$ and the untranslated part u calculated in the above step are compared. If the source phrase $\bar{f}$ does not include the untranslated part u and is not included u, the new candidate is rejected.

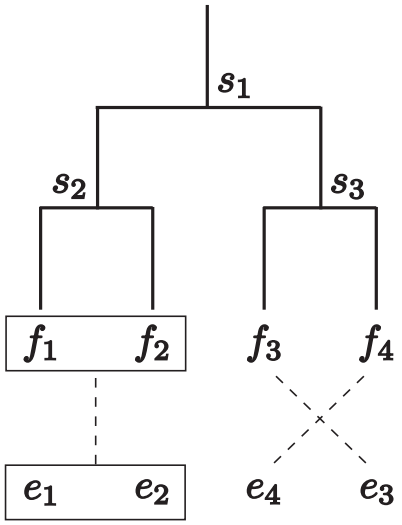

Figure 2.4: Example of a target candidate including a phrase.

3. In the accepted candidate, the reordering positions for all subtrees included the source side parse-tree are checked by comparing the source phrase $\bar{f}$ with the source phrase sequence used before.

Subtrees checked reordering positions are assigned a probability–monotone or swap–by the proposed reordering model, and the target word order is evaluated by Equation (2.1).

Phrase-based statistical machine translation uses a "phrase" as the translation unit. However, the proposed reordering model needs a "word" order. Because "word" alignments from the source phrase to target phrase are not clear, we cannot determine the reordering position of subtree included in a phrase. Therefore, in the decoding process using the proposed reordering model, we define that higher probability, monotone or swap, are assigned to subtrees included in a source phrase. For example, in Figure 2.4, the source sentence $[[f_1, f_2], f_3, f_4]$ is translated into the target sentence $[[e_1, e_2], e_4, e_3]$, where $[f_1, f_2]$ and $[e_1, e_2]$ are used as phrases. Then, the source phrase $[f_1, f_2]$ includes the subtree s_2. If the monotone probabilities of subtrees s_1, s_2, and s_3 are 0.8, 0.4 and 0.7, the proposed reordering probability is $0.8 \times 0.6 \times 0.3 = 0.144$. If a source phrase is $[f_1, f_2, f_3, f_4]$ and a source-side parse-tree has the same tree structure used in Figure 2.4, the subtrees s_1, s_2, and s_3 are assigned higher reordering probabilities. If the source phrase $[f_1, f_2, f_3, f_4]$ used in Figure 2.4, the subtrees s_1, s_2, and s_3 are assigned higher reordering probabilities.

Non-binary subtrees are often observed in the source-side parse-tree. When a source phrase $\bar{f}$ is included in a non-binary subtree and does not include a non-binary subtree, we cannot determine the reordering position. For example, the reordering position of subtree s_2 in Figure 2.5, which includes the phrase $[f_3, f_4]$, can not be determined. In this

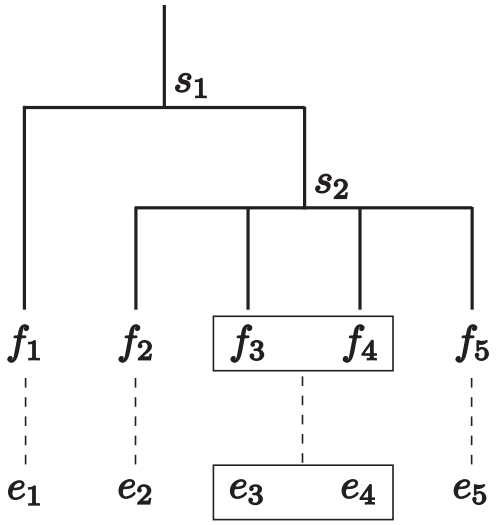

Figure 2.5: Example of a non-binary subtree including a phrase.

		English	Japanese
Train	Sentences	1.0M	
	Words	24.6M	28.8M
Dev	Sentences	2.0K	
	Words	50.1K	58.7K
Test	Sentences	2.0K	
	Words	49.5K	58.0K

Table 2.2: Statistics of training, development and test corpus for E-J translation.

case, we define that such subtrees are also to be assigned a higher probability.

2.3 Experiments

To evaluate the proposed model, we conducted two experiments: English-to-Japanese and English-to-Chinese translation.

2.3.1 English-to-Japanese Paper Abstract Translation Experiments

The first experiment was the English-to-Japanese (E-J) translation. Table 2.2 shows the training, development and test corpus statistics. JST Japanese-English paper abstract corpus consists of 1.0M parallel sentences were used for model training. This corpus

was constructed from 2.0M Japanese-English paper abstract corpus belongs to JST [31] by NICT using the method of Uchiyama and Isahara [32]. For phrase-based translation model training, we used the GIZA++ toolkit [29], and 1.0M bilingual sentences. For language model training, we used the SRI language model toolkit [33], and 1.0M sentences for the translation model training. The language model type was word 5-gram smoothed by Kneser-Ney discounting [34]. To tune the decoder parameters, we conducted minimum error rate training [35] with respect to the four word BLEU score [36] using 2.0K development sentence pairs. The test set with 2.0K sentences is used. In the evaluation and development sets, a single reference was used. For the creation of English sentence parse trees and segmentation of the English, we used the Charniak parser [37]. We used Chasen [38] for segmentation of the Japanese sentences. We used CleopATRa made at ATR for the decoding, which is compatible with Moses [39]. The performance of this decoder was configured to be the same as Moses. Other conditions were the same as the default conditions of the Moses decoder.

In this experiment, the following three methods were compared.

- Baseline : The IBM constraints and the lexical reordering model were used for target word reordering.
- IST-ITG : The IST-ITG constraints, the IBM constraints, and the lexical reordering model were used for target word reordering.
- Proposed : The proposed reordering model, the IBM constraints, and the lexical reordering model were used for target word reordering.

During minimum error training, each method used each reordering model and reordering constraint.

The proposed reordering model are trained from 1.0M bilingual sentences which are used for the translation model training. The amount of available training samples represented by subtrees was 9.8M. In the available training samples, there were 54K subtree types. The heuristic threshold was 10, and subtrees with training samples of less than 10 were clustered. The proposed reordering model consisted of 5,960 subtrees types and one clustered model. The models not including the clustered model covered 99.29% of all training samples.

The BLEU and WER are presented in Table 2.3. In comparing “Baseline” method with “IST-ITG” method, the improvement in BLEU was a 1.44-point and improvement in WER was 4.76%. Furthermore, in comparing “IST-ITG” method with “Proposed” method, the improvement in BLEU was a 0.49-point and improvement in WER was 0.65%. Table 2.4 shows the number of outputs that improved or got worse in BLEU after comparing

	Baseline	IST-ITG	Proposed
BLEU	27.87	29.31	29.80
WER	77.20	72.44	71.79

Table 2.3: BLEU score results for E-J translation. (1-reference)

	positive	negative	equal
# of outputs	605	539	851

Table 2.4: The number of output that "Proposed" improved and got worse in BLEU score from "IST-ITG" for E-J translation.

"Proposed" method with "IST-ITG" method. These results indicate a statistically significant difference at 95% confidence level between "Proposed" method and "IST-ITG" method. Both the IST-ITG constraints and the proposed reordering model fixed the phrase position for the global reorderings. However, the proposed method can conduct a probabilistic evaluation of target word reorderings which the IST-ITG constraints cannot. When the source sentence consists a few words (i.e. less than 15 words), the proposed reordering model obtains the similar performance with the IST-ITG constraints. However, when the source sentence consists many words and the source sentence structure is complex, the results using the proposed reordering model is better than one using the IST-ITG constraints. In this experiment, when the number of source words was more than 30, 45% of test sentences were improved by the proposed reordering model. Therefore, "Proposed" method resulted in a better BLEU and WER. The improvement could clearly be seen from visual inspection of the output, a few examples of which are presented in the Appendix.

2.3.2 NIST MT08 English-to-Chinese Translation Experiments

Next, we conducted English-to-Chinese (E-C) newspaper translation experiments for different language pairs. The NIST MT08 evaluation campaign English-to-Chinese translation track was used for the training and evaluation corpora. Table 2.5 shows the training, development and test corpus statistics. For the translation model training, we used 4.6M bilingual sentences. For the language model training, we used 4.6M sentences which are used for the translation model training. The language model type was word 3-gram smoothed by Kneser-Ney discounting. A development set with 1.6K sentences was used as evaluation data in the Chinese-to-English translation track for the NIST MT07 evaluation campaign. A single reference was used in the development set. The evaluation set with 1.9K sentences is the same as the MT08 evaluation data, with 4 references. In this

		English	Chinese
Train	Sentences	4.6M	
	Words	79.6M	73.4M
Dev	Sentences	1.6K	
	Words	46.4K	39.0K
Test	Sentences	1.9K	
	Words	45.7K	47.0K (Ave.)

Table 2.5: Statistics of training, development and test corpus for E-C translation.

	Baseline	IST-ITG	Proposed
BLEU	17.54	18.60	18.93
WER	78.07	75.43	75.57

Table 2.6: BLEU score results for E-C translation. (4-reference)

experiment, the compared methods were the same as in the E-J experiment.

The proposed reordering model are trained from 4.6M bilingual sentences which are used for the translation model training. The amount of available training samples represented by subtrees was 39.6M. In the available training samples, there were 193K subtree types. As in the E-J experiments, the heuristic threshold was 10. The proposed reordering model consisted of 18,955 subtree types and one clustered model. The models not including the clustered model covered 99.45% of all training samples.

The BLEU and WER are presented in Table 2.6. In comparing "Baseline" method with "IST-ITG" method, the improvement in BLEU was a 1.06-point. Furthermore, in comparing "IST-ITG" method with "Proposed" method, the improvement in BLEU was a 0.33-point. As in the E-J experiments, "Proposed" method performed the highest BLEU. Consequently, we demonstrated that the proposed method is effective for multiple language pairs. However, the improvement of BLEU and WER in E-C translation is smaller than the improvement in E-J translation. Table 2.7 shows the number of outputs that improved or got worse in BLEU after comparing "Proposed" method with "IST-ITG" method. These results cannot indicate a statistically significant difference at 95% confidence level between "Proposed" method and "IST-ITG" method. That is because English and Chinese are similar sentence tree structures, such as SVO-languages (Japanese is SOV-language). When the sentence tree structures are different, the proposed reordering model is effective.

	positive	negative	equal
# of outputs	463	428	968

Table 2.7: The number of output that "Proposed" improved and got worse in BLEU score from "IST-ITG" for E-C translation.

2.4 Summary

This chapter proposed a new word reordering model using a source-side parse-tree for phrase-based statistical machine translation. The proposed model is an extension of the IST-ITG constraints. In both IST-ITG constraints and the proposed method, the target-side word order is obtained by rotating nodes of the source-side tree structure. Both the IST-ITG constraints and the proposed reordering model fix the phrase position for the global reorderings. However, the proposed method can conduct a probabilistic evaluation of target word reorderings which the IST-ITG constraints cannot. In E-J and E-C translation experiments, the proposed method resulted in a 0.49-point improvement (29.31 to 29.80) and a 0.33-point improvement (18.60 to 18.93) in word BLEU-4 compared with IST-ITG constraints, respectively. This indicates the validity of the proposed reordering model.

Future work will focus on a simultaneous training of translation and reordering models. Moreover, we will deal with difference between source and target tree structures in multi level like in [40].

Chapter 3

Bayesian Context Clustering Using Cross Validation for Speech Recognition

In hidden Markov model (HMM) based speech recognition systems [6], accurate acoustic modeling is necessary for reducing recognition error rate. The maximum likelihood (ML) criterion is one of the standard criteria for training acoustic models in speech recognition. The ML criterion guarantees to estimate the true values of the parameters as the amount of training data infinitely increases. However, the performance of current speech recognition systems is still far from satisfactory. In a real environment, there are many fluctuations originating from various factors such as the speaker, speaking style, and noise. A mismatch between the training and testing conditions often brings a drastic degradation in performance. However, since the ML criterion produces a point estimate of model parameters, the estimation accuracy may be degraded due to the over-fitting problem when the amount of training data is insufficient.

On the other hand, the Bayesian approach considers the posterior distribution of all variables [7]. That is, all the variables introduced when models are parameterized, such as model parameters and latent variables, are regarded as random variables, and their posterior distributions are obtained based on the Bayes theorem. The difference between the Bayesian and ML approaches is that the target of estimation is the distribution function in the Bayesian approach whereas it is the parameter value in the ML approach. Based on this posterior distribution estimation, the Bayesian approach can generally achieve more robust model construction and classification than the ML approach [8–10]. However, the Bayesian approach requires complicated integral and expectation computations to obtain posterior distributions when models have latent variables. Since the acoustic

models used in speech recognition (e.g., HMMs) have the latent variables, it is difficult to apply the Bayesian approach to speech recognition directly with no approximation. Recently, the Variational Bayesian (VB) approach has been proposed in the field of learning theory to avoid complicated computations by employing the variational approximation technique [17]. With this VB approach, approximate posterior distributions are obtained effectively by iterative calculations similar to the Expectation-Maximization (EM) algorithm used in the ML approach. The VB approach has been applied to speech recognition and it shows good performance [11].

The VB approach has also been applied to the context clustering [11,12]. It is well known that contextual factors affect speech. Therefore, context-dependent acoustic models (e.g., triphone HMMs) are widely used in HMM-based speech recognition [41,42]. Although a large number of context-dependent acoustic models can capture variations in speech data, too many model parameters lead to the over-fitting problem. Consequently, maintaining a good balance between model complexity and the amount of training data is very important for obtaining high generalization performance. The decision tree based context clustering [43] is an efficient method for dealing with the problem of data sparseness, for both estimating robust model parameter of context-dependent acoustic models and obtaining predictive distributions of unseen contexts. This method constructs a model parameter tying structure which can assign a sufficient amount of training data to each HMM state. The tree is grown step by step, choosing questions that divide the set of contexts using a greedy strategy to maximize an objective function.

The ML criterion is inappropriate as a model selection criterion because it increases monotonically as the number of states increases. Some heuristic thresholding is therefore necessary to stop splitting nodes in the context clustering. To solve this problem, the minimum description length (MDL) criterion has been employed to select the model structure [44]. However, the MDL criterion is based on an asymptotic assumption, therefore it is ineffective when the amount of training data is small. On the other hand, the Bayesian information criterion (BIC) [45] has been proposed as an approximated Bayesian criterion. However, since the BIC is practically the same as the MDL criterion, The BIC is also ineffective when the amount of training data is small. In contrast to the BIC, the model selection based on the VB method has been proposed [11,12]. The VB method can select an appropriate model structure, even when there are insufficient amounts of data, because it does not use an asymptotic assumption. Therefore, the speech recognition framework which consistently applies the VB method is effective for estimating appropriate acoustic models and model structures.

The Bayesian approach has an advantage that it can utilize prior distributions which represent the prior information of model parameters. In the Bayesian approach, since prior dis-

tributions of model parameters affect the estimation of posterior distributions and model selection, the determination of prior distributions is an important problem for estimating appropriate acoustic models. As the determination technique of prior distributions, some techniques have been proposed in the filed of machine learning, e.g., using uninformative (uniform) prior distributions, hierarchical Bayesian methods, and empirical Bayesian methods [46]. However, it has not been thoroughly investigated in speech recognition, and the determination technique of prior distributions has not performed well. This chapter proposes a prior distribution determination technique using cross validation and applies it to the context clustering for the speech recognition framework based on Bayesian approach. The cross validation method is known as a straightforward and useful method for model structure optimization [13, 14]. The main idea behind cross validation is to split data for estimating the risk of each model. Part of data is used for training each model, and the remaining part is used for estimating the risk of the model. Then, the cross validation method selects the model with the smallest estimated risk. The cross validation method avoids the over-fitting problem because the training data is independent from the validation data. The context clustering based on the ML criterion using cross validation has been proposed, and it can select a more appropriate model structure than the conventional ML criterion [15]. The proposed method can be regarded as an extension of context clustering using cross validation to the Bayesian approach. Using prior distributions determined by the cross validation, it is expected that a higher generalization ability is achieved and an appropriate model structure can be selected in the context clustering without any tuning parameters.

The rest of the chapter is organized as follows. Section 3.1 describes speech recognition based on the variational Bayesian method. Section 3.2 derives the prior distribution determination technique using cross validation and apply it to the context clustering. Results of the continuous phoneme recognition experiments are shown in Section 3.3. Concluding remarks and future plans are presented in the final section.

3.1 Speech recognition based on variational Bayesian method

3.1.1 Bayesian approach

The output distribution is obtained based on a left-to-right HMM which has been widely used to represent an acoustic model for speech recognition. Let $\boldsymbol{O} = (\boldsymbol{o}_1, \boldsymbol{o}_2, \ldots, \boldsymbol{o}_T)$ be a set of training data of D dimensional feature vectors, and T is used to denote the

number of frames. The log output distribution is represented by

$$\begin{aligned}\log P(\boldsymbol{O}, \boldsymbol{Z} \mid \boldsymbol{\Lambda}) &= \sum_{i=1}^{N} Z_1^i \log \pi_i + \sum_{t=1}^{T-1} \sum_{i=1}^{N} \sum_{j=1}^{N} Z_t^i Z_{t+1}^j \log a_{ij} \\ &\quad + \sum_{t=1}^{T} \sum_{i=1}^{N} Z_t^i \log \mathcal{N}(\boldsymbol{o}_t \mid \boldsymbol{\mu}_i, \boldsymbol{S}_i^{-1}) \end{aligned} \tag{3.1}$$

where $\boldsymbol{Z} = (Z_1, Z_2, \ldots, Z_T)$ is a sequence of latent variables which represent HMM states, $Z_t \in \{1, \ldots, N\}$ denotes a state at frame t, and N is the number of states in an HMM.

$$Z_t^i = \delta(Z_t, i) = \begin{cases} 1 & \text{if } Z_t = i \\ 0 & \text{otherwise} \end{cases} \tag{3.2}$$

A set of model parameters $\boldsymbol{\Lambda} = \{\pi_i, a_{ij}, \boldsymbol{\mu}_i, \boldsymbol{S}_i\}_{i,j=1}^{N}$ consists of the initial state probability π_i of state i, the state transition probability a_{ij} from state i to state j, the mean vector $\boldsymbol{\mu}_i$ and the covariance matrix $\boldsymbol{S}_i^{-1}$ of a Gaussian distribution $\mathcal{N}(\cdot \mid \boldsymbol{\mu}_i, \boldsymbol{S}_i^{-1})$. [1]

In HMM-based speech recognition, the ML criterion has typically been used to train HMMs. In the ML criterion, the optimal model parameters are estimated by maximizing the likelihood for given training data as follows.

$$\begin{aligned}\boldsymbol{\Lambda}_{\mathrm{ML}} &= \arg\max_{\boldsymbol{\Lambda}} P(\boldsymbol{O} \mid \boldsymbol{\Lambda}) \\ &= \arg\max_{\boldsymbol{\Lambda}} \sum_{\boldsymbol{Z}} P(\boldsymbol{O}, \boldsymbol{Z} \mid \boldsymbol{\Lambda}) \end{aligned} \tag{3.3}$$

The model parameters can be estimated using an iterative procedure such as the EM algorithm [47] because it is difficult to obtain the model parameters $\boldsymbol{\Lambda}_{\mathrm{ML}}$ analytically. The ML criterion guarantees to estimate the true values of the model parameters as the amount of training data infinitely increases. However, the ML criterion produces a point estimate of model parameters. The use of point estimate will cause an over-fitting problem when the amount of training data is insufficient. A overfitted model will generally have poor predictive performance, because it captures minor fluctuations in the training data.

The Bayesian approach assumes that a set of model parameters $\boldsymbol{\Lambda}$ is random variables, while the ML approach estimates constant model parameters. The posterior distribution for a set of model parameters $\boldsymbol{\Lambda}$ is given by the famous Bayes theorem as follows.

$$P(\boldsymbol{\Lambda} \mid \boldsymbol{O}) = \frac{P(\boldsymbol{O} \mid \boldsymbol{\Lambda}) P(\boldsymbol{\Lambda})}{P(\boldsymbol{O})} \tag{3.4}$$

[1] Although a multi-mixture Gaussian is typically used as a state output probability distribution in recent HMM-based speech recognition systems, a single Gaussian is assumed as a state output probability distribution in this chapter for simplification.

where $P(\mathbf{\Lambda})$ is a prior distribution for $\mathbf{\Lambda}$, and $P(\boldsymbol{O})$ is an evidence.

Once the posterior distribution $P(\mathbf{\Lambda} \mid \boldsymbol{O})$ is estimated, the predictive distribution for input data $\boldsymbol{X}$ is represented by

$$P(\boldsymbol{X} \mid \boldsymbol{O}) = \int P(\boldsymbol{X} \mid \mathbf{\Lambda}) P(\mathbf{\Lambda} \mid \boldsymbol{O}) d\mathbf{\Lambda} \tag{3.5}$$

The model parameters are integrated out in Eq. (3.5) so that the effect of over-fitting is mitigated, and robust classification is achieved. However, the Bayesian approach requires complicated integral and expectation calculations to obtain posterior distributions when models include latent variables. To overcome this problem, maximum a posterior (MAP) approach has been proposed [48]. In the MAP approach, the optimal model parameters are estimated by maximizing the posterior probability. The MAP criterion can utilize the prior distribution $P(\mathbf{\Lambda})$, and can be seen as an extension of the ML criterion. However, it also produces a point estimate of HMM parameters. Consequently, it still has the effect of the over-fitting due to a point estimate.

On the other hand, the variational Bayesian (VB) method has been proposed as a tractable approximation method of the Bayesian approach [17]. The VB method avoids complicated computations by employing the variational approximation technique, and estimates approximate posterior distributions effectively by iterative calculations similar to the EM algorithm in the ML approach.

3.1.2 Variational Bayesian method

In the variational Bayesian method, an approximate posterior distribution is estimated by maximizing a lower bound of log marginal likelihood $\mathcal{F}$ instead of the true likelihood. A lower bound of log marginal likelihood is defined by using Jensen's inequality.

$$\begin{aligned}
\log P(\boldsymbol{O}) &= \log \sum_{\boldsymbol{Z}} \int P(\boldsymbol{O}, \boldsymbol{Z} \mid \mathbf{\Lambda}) P(\mathbf{\Lambda}) d\mathbf{\Lambda} \\
&= \log \sum_{\boldsymbol{Z}} \int Q(\boldsymbol{Z}, \mathbf{\Lambda}) \frac{P(\boldsymbol{O}, \boldsymbol{Z} \mid \mathbf{\Lambda}) P(\mathbf{\Lambda})}{Q(\boldsymbol{Z}, \mathbf{\Lambda})} d\mathbf{\Lambda} \\
&\geq \sum_{\boldsymbol{Z}} \int Q(\boldsymbol{Z}, \mathbf{\Lambda}) \log \frac{P(\boldsymbol{O}, \boldsymbol{Z} \mid \mathbf{\Lambda}) P(\mathbf{\Lambda})}{Q(\boldsymbol{Z}, \mathbf{\Lambda})} d\mathbf{\Lambda} \\
&= \mathcal{F}
\end{aligned} \tag{3.6}$$

where $Q(\boldsymbol{Z}, \mathbf{\Lambda})$ is an arbitrary distribution. The relation between the log marginal likelihood and the lower bound $\mathcal{F}$ is represented by using the Kullback-Leibler (KL) diver-

gence [49] between $Q(\boldsymbol{Z}, \boldsymbol{\Lambda})$ and true posterior distribution $P(\boldsymbol{Z}, \boldsymbol{\Lambda} \mid \boldsymbol{O})$.

$$\begin{aligned} \log P(\boldsymbol{O}) - \mathcal{F} &= \mathrm{KL}[Q(\boldsymbol{Z}, \boldsymbol{\Lambda}) \mid P(\boldsymbol{Z}, \boldsymbol{\Lambda} \mid \boldsymbol{O})] \\ &= \sum_{\boldsymbol{Z}} \int Q(\boldsymbol{Z}, \boldsymbol{\Lambda}) \log \frac{Q(\boldsymbol{Z}, \boldsymbol{\Lambda})}{P(\boldsymbol{Z}, \boldsymbol{\Lambda} \mid \boldsymbol{O})} d\boldsymbol{\Lambda} \end{aligned} \tag{3.7}$$

where $\mathrm{KL}[Q(\boldsymbol{Z}, \boldsymbol{\Lambda}) \mid P(\boldsymbol{Z}, \boldsymbol{\Lambda} \mid \boldsymbol{O})]$ denote a KL divergence. As the difference between the true log marginal likelihood and the lower bound is reduced, $Q(\boldsymbol{Z}, \boldsymbol{\Lambda})$ approximate the true posterior distribution $P(\boldsymbol{Z}, \boldsymbol{\Lambda} \mid \boldsymbol{O})$. Therefore, the optimal posterior distribution is estimated by the variational method, which results in minimizing the right hand side of Eq. (3.7)

To obtain approximate posterior distributions (VB posterior distributions) $Q(\boldsymbol{Z}, \boldsymbol{\Lambda})$, it is assumed that random variables are conditionally independent each other.

$$Q(\boldsymbol{Z}, \boldsymbol{\Lambda}) = Q(\boldsymbol{Z}) Q(\boldsymbol{\Lambda}) \tag{3.8}$$

Under this assumption, the optimal VB posterior distributions which maximize the objective function $\mathcal{F}$ are given by the variational method as follows.

$$Q(\boldsymbol{\Lambda}) = C_{\boldsymbol{\Lambda}} P(\boldsymbol{\Lambda}) \exp\left\{ \langle \log P(\boldsymbol{O}, \boldsymbol{Z} \mid \boldsymbol{\Lambda}) \rangle_{Q(\boldsymbol{Z})} \right\} \tag{3.9}$$

$$Q(\boldsymbol{Z}) = C_{\boldsymbol{Z}} \exp\left\{ \langle \log P(\boldsymbol{O}, \boldsymbol{Z} \mid \boldsymbol{\Lambda}) \rangle_{Q(\boldsymbol{\Lambda})} \right\} \tag{3.10}$$

where $\langle \cdot \rangle_Q$ denotes the expectation with respect to Q, $C_{\boldsymbol{\Lambda}}$ and $C_{\boldsymbol{Z}}$ are the normalization terms of $Q(\boldsymbol{\Lambda})$ and $Q(\boldsymbol{Z})$, respectively. Moreover, it is assumed that the model parameters $\boldsymbol{\pi} = \{\pi_i\}_{i=1}^N$, $\boldsymbol{a}_i = \{a_{ij}\}_{j=1}^N$, and $\{\boldsymbol{\mu}_i, \boldsymbol{S}_i\}_{i=1}^N$ are independent each other in the prior distribution $P(\boldsymbol{\Lambda})$. Therefore, the prior distribution $P(\boldsymbol{\Lambda})$ can be represented as follows.

$$P(\boldsymbol{\Lambda}) = P(\boldsymbol{\pi}) \prod_{i=1}^N P(\boldsymbol{a}_i) \prod_{i=1}^N P(\boldsymbol{\mu}_i, \boldsymbol{S}_i) \tag{3.11}$$

By using this assumption, the posterior distribution $Q(\boldsymbol{\Lambda})$ and its normalization term $C_{\boldsymbol{\Lambda}}$ can be written as follows.

$$Q(\boldsymbol{\Lambda}) = Q(\boldsymbol{\pi}) \prod_{i=1}^N Q(\boldsymbol{a}_i) \prod_{i=1}^N Q(\boldsymbol{\mu}_i, \boldsymbol{S}_i) \tag{3.12}$$

$$C_{\boldsymbol{\Lambda}} = C_{\boldsymbol{\pi}} \prod_{i=1}^N C_{\boldsymbol{a}_i} \prod_{i=1}^N C_{\boldsymbol{\mu}_i, \boldsymbol{S}_i} \tag{3.13}$$

From Eqs. (3.1), (3.2) and (3.9)–(3.13), the posterior distributions of model parameters

are given as follows.

$$
\begin{aligned}
Q(\boldsymbol{\pi}) &= C_{\boldsymbol{\pi}} P(\boldsymbol{\pi}) \exp\left\{ \sum_{i=1}^{N} \langle Z_1^i \rangle \log \pi_i \right\} && (3.14) \\
Q(\boldsymbol{a}_i) &= C_{\boldsymbol{a}_i} P(\boldsymbol{a}_i) \\
&\quad \times \exp\left\{ \sum_{j=1}^{N} \sum_{t=1}^{T-1} \langle Z_t^i Z_{t+1}^j \rangle \log a_{ij} \right\} && (3.15) \\
Q(\boldsymbol{\mu}_i, \boldsymbol{S}_i) &= C_{\boldsymbol{\mu}_i, \boldsymbol{S}_i} P(\boldsymbol{\mu}_i, \boldsymbol{S}_i) \\
&\quad \times \exp\left\{ \sum_{t=1}^{T} \langle Z_t^i \rangle \log \mathcal{N}(\boldsymbol{o}_t \mid \boldsymbol{\mu}_i, \boldsymbol{S}_i) \right\} && (3.16)
\end{aligned}
$$

where $\boldsymbol{\pi} = \{\pi_i\}_{i=1}^N$ is a set of initial state probabilities, $\boldsymbol{a}_i = \{a_{ij}\}_{j=1}^N$ is a set of state transition probabilities from state i, and $\langle Z_t^i \rangle$ and $\langle Z_t^i Z_{t+1}^j \rangle$ are the expectation value with respect to $Q(\boldsymbol{Z})$ as follows.

$$
\begin{aligned}
\langle Z_t^i \rangle &= \sum_{\boldsymbol{Z}} Q(\boldsymbol{Z}) Z_t^i && (3.17) \\
\langle Z_t^i Z_{t+1}^j \rangle &= \sum_{\boldsymbol{Z}} Q(\boldsymbol{Z}) Z_t^i Z_{t+1}^j && (3.18)
\end{aligned}
$$

The posterior distribution $Q(\boldsymbol{Z})$ can be represented by using Eqs. (3.1), (3.2) and (3.10)–(3.16) as follows.

$$
\begin{aligned}
Q(\boldsymbol{Z}) &= C_{\boldsymbol{Z}} \prod_{i=1}^{N} \exp\left\{ Z_1^i \left\langle \log \pi_i \right\rangle_{Q(\boldsymbol{\pi})} \right\} \\
&\quad \times \prod_{t=1}^{\hat{T}-1} \prod_{i=1}^{N} \prod_{j=1}^{N} \exp\left\{ Z_t^i Z_{t+1}^j \left\langle \log a_{ij} \right\rangle_{Q(\boldsymbol{a}_i)} \right\} \\
&\quad \times \prod_{t=1}^{\hat{T}} \prod_{i=1}^{N} \exp\left\{ Z_t^i \left\langle \log \mathcal{N}(\boldsymbol{o}_t \mid \boldsymbol{\mu}_i, \boldsymbol{S}_i^{-1}) \right\rangle_{Q(\boldsymbol{\mu}_i, \boldsymbol{S}_i)} \right\}
\end{aligned} \tag{3.19}
$$

The posterior distribution $Q(\boldsymbol{Z})$ is similar to the likelihood function of an HMM when the terms $\exp\left\{ \left\langle \log \pi_i \right\rangle_{Q(\boldsymbol{\pi})} \right\}$, $\exp\left\{ \left\langle \log a_{ij} \right\rangle_{Q(\boldsymbol{a}_i)} \right\}$, and $\exp\left\{ \left\langle \log \mathcal{N}(\boldsymbol{x}_t \mid \boldsymbol{\mu}_i, \boldsymbol{S}_i^{-1}) \right\rangle_{Q(\boldsymbol{\mu}_i, \boldsymbol{S}_i)} \right\}$ are respectively used as the initial state probability of state i, the state transition probability from state i to state j, and the output probability of state i. Therefore, Eqs. (3.17) and (3.18) can be computed efficiently by the Forward-Backward algorithm.

3.1.3 Prior distribution

In the Bayesian approach, a conjugate prior distribution is widely used as a prior distribution. Prior distributions are respectively represented as follows.

$$P(\boldsymbol{\pi}) = \mathcal{D}(\{\pi_i\}_{i=1}^N \mid \{\phi_i\}_{i=1}^N), \tag{3.20}$$

$$P(\boldsymbol{a}_i) = \mathcal{D}(\{a_{ij}\}_{j=1}^N \mid \{\alpha_{ij}\}_{j=1}^N), \tag{3.21}$$

$$P(\boldsymbol{\mu}_i, \boldsymbol{S}_i) = \mathcal{N}(\boldsymbol{\mu}_i \mid \boldsymbol{\nu}_i, (\xi_i \boldsymbol{S}_i)^{-1})\, \mathcal{W}(\boldsymbol{S}_i \mid \eta_i, \boldsymbol{B}_i) \tag{3.22}$$

where $\mathcal{D}(\cdot)$ is a Dirichlet distribution, and $\mathcal{N}(\cdot)\mathcal{W}(\cdot)$ is a Gauss-Wishart distribution. Moreover, $\{\phi_i, \alpha_{ij}, \xi_i, \eta_i,$
$\boldsymbol{\nu}_i, \boldsymbol{B}_i\}_{i,j=1}^N$ is a set of hyper-parameters. When these conjugate prior distributions are used, the posterior distributions are represented by the same set of parameters $\left\{\bar{\phi}_i, \bar{\alpha}_{ij}, \bar{\xi}_i, \bar{\eta}_i, \bar{\boldsymbol{\nu}}_i, \bar{\boldsymbol{B}}_i\right\}_{i,j=1}^N$.

3.1.4 Update of posterior distribution

The posterior distribution of model parameters $Q(\boldsymbol{\Lambda})$ can be updated by sufficient statistics of the training data as follows.

$$\bar{\phi}_i = \phi_i + \langle Z_1^i \rangle \tag{3.23}$$

$$\bar{\alpha}_{ij} = \alpha_{ij} + \bar{T}_{ij} \tag{3.24}$$

$$\bar{\xi}_i = \xi_i + \bar{T}_i \tag{3.25}$$

$$\bar{\eta}_i = \eta_i + \bar{T}_i \tag{3.26}$$

$$\bar{\boldsymbol{\nu}}_i = \frac{\bar{T}_i \bar{\boldsymbol{o}}_i + \xi_i \boldsymbol{\nu}_i}{\bar{T}_i + \xi_i} \tag{3.27}$$

$$\bar{\boldsymbol{B}}_i = \bar{T}_i \bar{\boldsymbol{C}}_i + \boldsymbol{B}_i + \frac{\bar{T}_i \xi_i}{\bar{T}_i + \xi_i} (\bar{\boldsymbol{o}}_i - \boldsymbol{\nu}_i)(\bar{\boldsymbol{o}}_i - \boldsymbol{\nu}_i)^\top \tag{3.28}$$

where the sufficient statistics $\bar{T}_i, \bar{T}_{ij}, \bar{\boldsymbol{o}}_i$ and $\bar{\boldsymbol{C}}_i$ are represented as follows.

$$\bar{T}_i = \sum_{t=1}^{T} \langle Z_t^i \rangle \tag{3.29}$$

$$\bar{T}_{ij} = \sum_{t=1}^{T-1} \langle Z_t^i Z_{t+1}^j \rangle \tag{3.30}$$

$$\bar{\boldsymbol{o}}_i = \frac{1}{\bar{T}_i} \sum_{t=1}^{T} \langle Z_t^i \rangle \boldsymbol{o}_t \tag{3.31}$$

$$\bar{\boldsymbol{C}}_i = \frac{1}{\bar{T}_i} \sum_{t=1}^{T} \langle Z_t^i \rangle (\boldsymbol{o}_t - \bar{\boldsymbol{o}}_i)(\boldsymbol{o}_t - \bar{\boldsymbol{o}}_i)^\top \tag{3.32}$$

These optimizations can be performed effectively by iterative calculations as the EM algorithm, which increases the value of objective function $\mathcal{F}$ at each iteration until convergence.

3.1.5 Speech recognition based on Bayesian approach

In the speech recognition based on the Bayesian approach, the test data $\boldsymbol{X} = (\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_{\hat{T}})$ are recognized by using the predictive distribution as follows.

$$\begin{aligned} \hat{\boldsymbol{H}} &= \arg\max_{\boldsymbol{H}} P(\boldsymbol{H} \mid \boldsymbol{X}, \boldsymbol{O}) \\ &= \arg\max_{\boldsymbol{H}} P(\boldsymbol{X} \mid \boldsymbol{O}, \boldsymbol{H}) P(\boldsymbol{H}) \end{aligned} \tag{3.33}$$

where $\boldsymbol{H}$ is a hypothesis of a phoneme sequence. The acoustic likelihood $P(\boldsymbol{X} \mid \boldsymbol{O}, \boldsymbol{H})$ can be approximated by the variational Bayesian method as model training described in Section 3.1.2.

$$\begin{aligned} \log P(\boldsymbol{X} \mid \boldsymbol{O}, \boldsymbol{H}) &= \log \sum_{\hat{\boldsymbol{Z}}} \int P(\boldsymbol{X}, \hat{\boldsymbol{Z}} \mid \boldsymbol{\Lambda}, \boldsymbol{H}) P(\boldsymbol{\Lambda} \mid \boldsymbol{O}) d\boldsymbol{\Lambda} \\ &\geq \sum_{\hat{\boldsymbol{Z}}} \int \hat{Q}(\hat{\boldsymbol{Z}}, \boldsymbol{\Lambda}) \log \frac{P(\boldsymbol{X}, \hat{\boldsymbol{Z}} \mid \boldsymbol{\Lambda}, \boldsymbol{H}) P(\boldsymbol{\Lambda} \mid \boldsymbol{O})}{\hat{Q}(\hat{\boldsymbol{Z}}, \boldsymbol{\Lambda})} d\boldsymbol{\Lambda} \\ &= \hat{\mathcal{F}}(\boldsymbol{X} \mid \boldsymbol{O}, \boldsymbol{H}) \end{aligned} \tag{3.34}$$

where $\hat{\boldsymbol{Z}} = (\hat{Z}_1, \hat{Z}_2, \ldots, \hat{Z}_{\hat{T}})$ is a sequence of HMM states for the test data $\boldsymbol{X}$, and $\hat{Q}(\hat{\boldsymbol{Z}}, \boldsymbol{\Lambda})$ is the VB posterior distribution which approximates the true posterior distribution $P(\hat{\boldsymbol{Z}}, \boldsymbol{\Lambda} \mid \boldsymbol{X})$. In the recognition process, the VB posterior distribution of model parameters $Q(\boldsymbol{\Lambda})$ estimated in the training part is used instead of $P(\boldsymbol{\Lambda} \mid \boldsymbol{O})$, and the same assumption as Eq. (3.8) is used. Moreover, it is assumed that the amount of test data is much smaller than the one of training data in this chapter. Then, the VB posterior distribution $\hat{Q}(\boldsymbol{\Lambda})$ is approximated by $Q(\boldsymbol{\Lambda})$. Therefore, the lower bound $\hat{\mathcal{F}}(\boldsymbol{X} \mid \boldsymbol{O}, \boldsymbol{H})$ is calculated by using $Q(\boldsymbol{\Lambda})$.

$$\begin{aligned} \hat{\mathcal{F}}(\boldsymbol{X} \mid \boldsymbol{O}, \boldsymbol{H}) &= \log \sum_{\hat{\boldsymbol{Z}}} \Bigg\{ \prod_{i=1}^{N} \exp\left\{ \hat{Z}_1^i \left\langle \log \pi_i \right\rangle_{Q(\boldsymbol{\pi})} \right\} \\ &\quad \times \prod_{t=1}^{\hat{T}-1} \prod_{i=1}^{N} \prod_{j=1}^{N} \exp\left\{ \hat{Z}_t^i \hat{Z}_{t+1}^j \left\langle \log a_{ij} \right\rangle_{Q(\boldsymbol{a}_i)} \right\} \\ &\quad \times \prod_{t=1}^{\hat{T}} \prod_{i=1}^{N} \exp\left\{ \hat{Z}_t^i \left\langle \log \mathcal{N}(\boldsymbol{x}_t \mid \boldsymbol{\mu}_i, \boldsymbol{S}_i^{-1}) \right\rangle_{Q(\boldsymbol{\mu}_i, \boldsymbol{S}_i)} \right\} \Bigg\} \end{aligned} \tag{3.35}$$

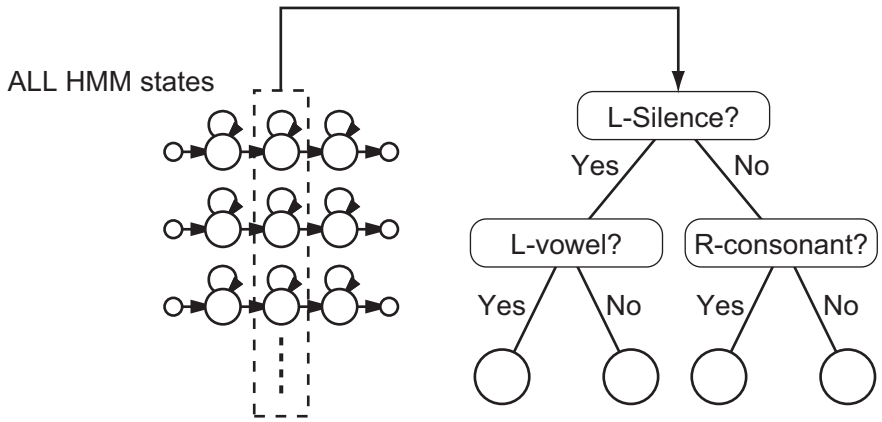

Figure 3.1: Overview of decision tree based context clustering.

Then, $\hat{\mathcal{F}}(\boldsymbol{X} \mid \boldsymbol{O}, \boldsymbol{H})$ is similar to the likelihood function of an HMM as Eq. (3.19). Although the accurate $\hat{\mathcal{F}}(\boldsymbol{X} \mid \boldsymbol{O}, \boldsymbol{H})$ is computed by considering all possible sequences of HMM states $\hat{\boldsymbol{Z}}$ as the training part, the Viterbi algorithm is applied in decoding as the ML approach.

3.2 Bayesian context clustering using cross validation

3.2.1 Bayesian context clustering

The decision tree based context clustering is a top-down clustering method to optimize the state tying structure for robust model parameter estimation [43]. A leaf node of the decision tree corresponds to a set of HMM states to be tied. The decision tree growing process begins with a root node that may have all HMM states, or all states associated with a particular phone, etc. Then, a question which divides the set of states into two subsets assigned respectively to two child nodes, "Yes" node and "No" node as illustrated in Fig. 3.1, is chosen so that the corresponding new HMM has the largest value of an objective function for training data. The decision tree is grown in a greedy fashion, successively splitting nodes by selecting the pair of a question and node that maximizes the gain of the objective function at each step.

In the Bayesian approach, an optimal model structure can be selected by maximizing the objective function $\mathcal{F}$. When a node is split into two nodes by the question q, the gain $\Delta\mathcal{F}_q$ is defined as the difference of $\mathcal{F}$ before and after splitting.

$$\Delta\mathcal{F}_q = \mathcal{F}_q^y + \mathcal{F}_q^n - \mathcal{F}_q^p \tag{3.36}$$

where $\mathcal{F}_q^y$ and $\mathcal{F}_q^n$ are the value of objective function $\mathcal{F}$ of split nodes by a question q, and

$\mathcal{F}_q^p$ is the value before a splitting. The question $\hat{q}$ for splitting a node is chosen from the question set as follows.

$$\hat{q} = \arg\max_q \Delta\mathcal{F}_q \tag{3.37}$$

By splitting nodes until $\Delta\mathcal{F}_{\hat{q}} \leq 0$, the decision tree that maximizes the objective function $\mathcal{F}$ is obtained.

In the decision tree based context clustering, it is typically assumed that the state occupancies are not changed by the split nodes. Then, the objective function $\mathcal{F}$ can be computed as follows.

$$\begin{aligned}\mathcal{F} &= -\log C_{\boldsymbol{\Lambda}} - \langle \log Q(\boldsymbol{Z}) \rangle_{Q(\boldsymbol{Z})} \\ &= -\sum_{i=1}^{N} \log C_{\boldsymbol{\mu}_i, \boldsymbol{S}_i} + \text{Const}\end{aligned} \tag{3.38}$$

From Eq. (3.38), the gain of the objective function $\Delta\mathcal{F}_q$ can be computed by the normalization term of the posterior distribution $C_{\boldsymbol{\mu}_i, \boldsymbol{S}_i}$. The normalization term $C_{\boldsymbol{\mu}_i, \boldsymbol{S}_i}$ is defined as follows.

$$\log C_{\boldsymbol{\mu}_i, \boldsymbol{S}_i} = \log \frac{\bar{C}_{\mathcal{N}_i} \bar{C}_{\mathcal{W}_i}}{C_{\mathcal{N}_i} C_{\mathcal{W}_i}} (2\pi)^{\frac{\bar{N}D}{2}} \tag{3.39}$$

where $C_{\mathcal{N}_i}$ and $C_{\mathcal{W}_i}$ denote the normalization terms of prior Gauss-Wishart distribution.

$$C_{\mathcal{N}_i} = (2\pi)^{-\frac{D}{2}} \xi_i^{\frac{D}{2}} \tag{3.40}$$

$$C_{\mathcal{W}_i} = \frac{|B_i|^{\frac{\eta_i}{2}}}{2^{\frac{\eta_i D}{2}} \pi^{\frac{D(D-1)}{4}} \prod_{j=1}^{D} \Gamma(\frac{\eta+1-j}{2})} \tag{3.41}$$

where $\Gamma(\cdot)$ is the Gamma function. The normalization terms of posterior Gauss-Wishart distribution are also denoted by $\bar{C}_{\mathcal{N}_i}$ and $\bar{C}_{\mathcal{W}_i}$, and they are represented by using posterior hyper-parameters $\bar{\xi}_i, \bar{\eta}_i$, and $\bar{\boldsymbol{B}}_i$ instead of prior hyper-parameters ξ_i, η_i, and $\boldsymbol{B}_i$ in Eqs. (3.40) and (3.41), respectively. The posterior hyper-parameters $\bar{\xi}_i, \bar{\eta}_i$, and $\bar{\boldsymbol{B}}_i$ can be calculated by using equations described in Section 3.1.4. From Eqs. (3.38)–(3.41), $\mathcal{F}$ can be computed by using the prior and posterior hyper-parameters. Since it is assumed that the state occupancies are not changed in the context clustering and the posterior hyper-parameters can be represented by using sufficient statistics and the prior hyper-parameters, the prior hyper-parameters are important parameters for the Bayesian context clustering.

If we have prior data $\tilde{\boldsymbol{O}}$ which is obtained from similar conditions (e.g., speaker, domain, recording condition) as the training data, the prior distribution can be constructed as $P(\boldsymbol{\Lambda}) = P(\boldsymbol{\Lambda} \mid \tilde{\boldsymbol{O}})$. When the prior data is given, the prior distribution is obtained by using the same approximation techniques as the variational Bayesian method described

in Section 3.1.2.

$$
\begin{aligned}
P(\boldsymbol{\Lambda} \mid \tilde{\boldsymbol{O}}) &\approx \tilde{Q}(\boldsymbol{\Lambda}) \\
&= \tilde{C}_{\boldsymbol{\Lambda}} \tilde{P}(\boldsymbol{\Lambda}) \exp\left\{ \left\langle \log P(\tilde{\boldsymbol{O}}, \tilde{\boldsymbol{Z}} \mid \boldsymbol{\Lambda}) \right\rangle_{\tilde{Q}(\tilde{\boldsymbol{Z}})} \right\}
\end{aligned}
\tag{3.42}
$$

where $\tilde{\boldsymbol{Z}}$ is a sequence of latent variables, and $\tilde{Q}(\tilde{\boldsymbol{Z}})$ is an approximate distribution of $P(\tilde{\boldsymbol{Z}} \mid \tilde{\boldsymbol{O}}, \boldsymbol{\Lambda})$. Although Eq. (3.42) still includes prior of prior distribution $\tilde{P}(\boldsymbol{\Lambda})$, we assumed that the prior of prior distribution $\tilde{P}(\boldsymbol{\Lambda})$ is a uniform distribution before the prior data is given. Then, prior distribution $P(\boldsymbol{\Lambda} \mid \tilde{\boldsymbol{O}})$ can be obtained as follows.

$$
\begin{aligned}
P(\boldsymbol{\Lambda} \mid \tilde{\boldsymbol{O}}) &\approx \tilde{C}_{\boldsymbol{\Lambda}} \exp\left\{ \left\langle \log P(\tilde{\boldsymbol{O}}, \tilde{\boldsymbol{Z}} \mid \boldsymbol{\Lambda}) \right\rangle_{\tilde{Q}(\tilde{\boldsymbol{Z}})} \right\} \\
&= \mathcal{D}(\{\pi_i\}_{i=1}^N \mid \{\tilde{T}_{0i}\}_{i=1}^N) \\
&\quad \times \prod_{i=1}^N \mathcal{D}(\{a_{ij}\}_{j=1}^N \mid \{\tilde{T}_{ij}\}_{j=1}^N) \\
&\quad \times \prod_{i=1}^N \left\{ \mathcal{N}(\boldsymbol{\mu}_i \mid \tilde{\boldsymbol{o}}_i, (\tilde{T}_i \boldsymbol{S}_i)^{-1}) \right. \\
&\quad \left. \times \mathcal{W}(\boldsymbol{S}_i \mid \tilde{T}_i + D, (\tilde{T}_i \boldsymbol{C}_i)) \right\}
\end{aligned}
\tag{3.43}
$$

The distribution $\tilde{Q}(\tilde{\boldsymbol{Z}})$ can be estimated via the EM algorithm using prior data $\tilde{\boldsymbol{O}}$. Statistics $\tilde{T}_{0i}$, $\tilde{T}_{ij}$ and $\tilde{T}_i$ denote the occupancy probabilities of initial state i, state transition from i to j, and state i with respect to the prior data, respectively. Moreover, $\tilde{\boldsymbol{o}}_i$ and $\tilde{\boldsymbol{C}}_i$ denote the mean vector and the covariance matrix of prior data in the i-th state, respectively. Thus, the prior distribution can be determined by sufficient statistics of the prior data. However, prior distributions are heuristically determined in many cases, because the prior data is not usually given in HMM-based speech recognition. Hyper-parameters affect the model selection as tuning parameters, therefore a determination technique of prior distributions is required to automatically select an appropriate model structure. One possible approach is to optimize the hyper-parameters so as to maximize the marginal likelihood of training data, as like the empirical Bayesian method [46]. However, it still needs tuning parameters which control influences of prior distributions, and often leads to the over-fitting problem as the ML criterion. In this chapter, we propose the prior distribution determination technique using cross validation and apply it to the context clustering.

3.2.2 Bayesian approach using cross validation

The cross validation method is a popular strategy for model selection [13, 14]. The main idea behind cross validation is to split data for estimating the risk of each model. Part

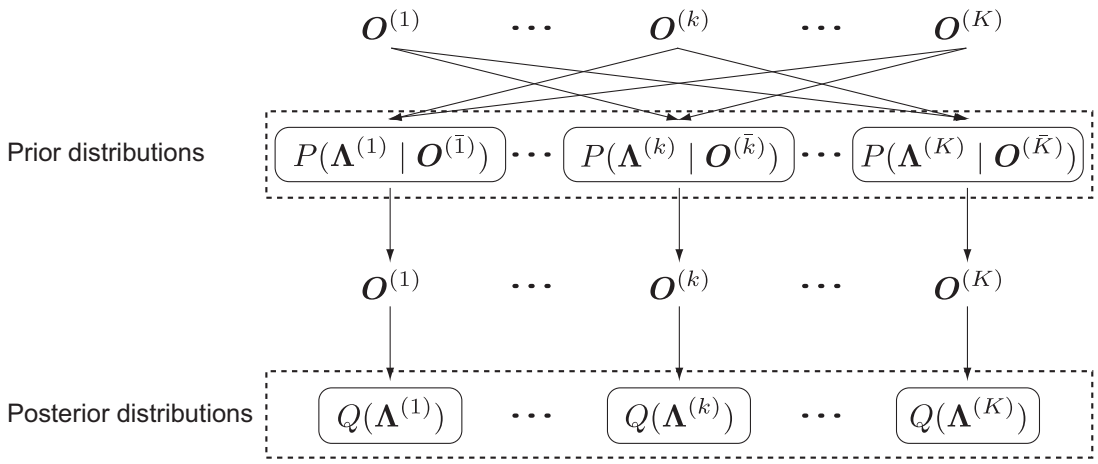

Figure 3.2: Overview of Bayesian approach using cross validation.

of data is used for training each model, and the remaining part is used for estimating the risk of the model. Then, the cross validation method selects the model with the smallest estimated risk. The basic form of cross validation is K-fold cross validation. In the K-fold cross validation method, the training data is randomly divided into K different groups. Then, a model is trained using $K-1$ groups of data, and the objective function is computed for the group excluded in the training. This process is repeated for K times with different combinations of $K-1$ groups. The value of objective function is accumulated and the accumulated value is used for evaluation of model structure.

In the Bayesian approach using K-fold cross validation, the training data $\boldsymbol{O}$ is divided at random into K subsets of training data $\left\{\boldsymbol{O}^{(1)}, \boldsymbol{O}^{(2)}, \ldots, \boldsymbol{O}^{(K)}\right\}$. For the k-th evaluation, $\boldsymbol{O}^{(\bar{k})} = \left\{\boldsymbol{O}^{(j)} \mid j \neq k\right\}$ and $\boldsymbol{O}^{(k)}$ are respectively used for the determination of prior distributions and the estimation of posterior distributions, i.e., $\boldsymbol{O}^{(\bar{k})}$ and $\boldsymbol{O}^{(k)}$ are used as prior data and training data. The Bayesian approach using cross validation considers the log marginal likelihood $\log P(\boldsymbol{O}^{(k)} \mid \boldsymbol{O}^{(\bar{k})})$. Using Jensen's inequality, the lower bound of log marginal likelihood $\mathcal{F}^{(k)}$ is defined as Eq. (3.6).

$$\log P(\boldsymbol{O}^{(k)} \mid \boldsymbol{O}^{(\bar{k})}) \geq \mathcal{F}^{(k)} \tag{3.44}$$

The optimal VB posterior distributions of model parameters are given by maximizing $\mathcal{F}^{(k)}$ with the variational method as Eq. (3.9).

$$\begin{aligned} Q(\boldsymbol{\Lambda}^{(k)}) &= C_{\boldsymbol{\Lambda}^{(k)}} P(\boldsymbol{\Lambda}^{(k)} \mid \boldsymbol{O}^{(\bar{k})}) \\ &\quad \times \exp\left\{\left\langle \log P(\boldsymbol{O}^{(k)}, \boldsymbol{Z}^{(k)} \mid \boldsymbol{\Lambda}^{(k)})\right\rangle_{Q(\boldsymbol{Z}^{(k)})}\right\} \end{aligned} \tag{3.45}$$

where $C_{\boldsymbol{\Lambda}^{(k)}}$ is a normalization term of $Q(\boldsymbol{\Lambda}^{(k)})$ and $P(\boldsymbol{\Lambda}^{(k)} \mid \boldsymbol{O}^{(\bar{k})})$ is a prior distribution

of the k-th cross validation which represents prior data $\boldsymbol{O}^{(\bar{k})}$. Figure 3.2 is an overview of the Bayesian approach using cross validation.

The cross validation method can select robust model structures because the objective value is calculated by evaluating open data. The Bayesian approach obtains robust predictive distributions and selects robust model structures while taking account of the amount of training data because posterior distributions of model parameters are used. Consequently, the Bayesian approach using cross validation can select model structures while taking account of the uncertainty of the data variables and model parameters, and the robustness can be improved from the standard Bayesian approach.

3.2.3 Bayesian context clustering using cross validation

The objective function $\mathcal{F}^{(\mathrm{CV})}$ is used in the Bayesian context clustering using cross validation. It is obtained by summing $\mathcal{F}^{(k)}$ for each fold.

$$\mathcal{F}^{(\mathrm{CV})} = \sum_{k=1}^{K} \mathcal{F}^{(k)} \tag{3.46}$$

In the proposed method, an optimal model structure can be selected by maximizing the objective function $\mathcal{F}^{(\mathrm{CV})}$. The question $\tilde{q}$ for splitting a node is chosen from the question set as Eq. (3.37).

$$\tilde{q} = \arg\max_{q} \Delta\mathcal{F}_{q}^{(\mathrm{CV})} \tag{3.47}$$

where $\Delta\mathcal{F}_{q}^{(\mathrm{CV})}$ is the gain in the value of the objective function $\mathcal{F}^{(\mathrm{CV})}$ when a node is split by the question q. The gain $\Delta\mathcal{F}_{q}^{(\mathrm{CV})}$ is obtained by

$$\Delta\mathcal{F}_{q}^{(\mathrm{CV})} = \mathcal{F}_{q}^{(\mathrm{CV})^{y}} + \mathcal{F}_{q}^{(\mathrm{CV})^{n}} - \mathcal{F}_{q}^{(\mathrm{CV})^{p}} \tag{3.48}$$

By splitting nodes until $\Delta\mathcal{F}_{\tilde{q}}^{(\mathrm{CV})} \leq 0$, the decision tree that maximizes the objective function $\mathcal{F}^{(\mathrm{CV})}$ is obtained.

The prior distribution of the k-th cross validation $P(\boldsymbol{\mu}^{(k)}, \boldsymbol{S}^{(k)} \mid \boldsymbol{O}^{(\bar{k})})$ is obtained from Eq. (3.43).

$$\begin{aligned} P(\boldsymbol{\mu}^{(k)}, \boldsymbol{S}^{(k)} \mid \boldsymbol{O}^{(\bar{k})}) &= \mathcal{N}(\boldsymbol{\mu}^{(k)} \mid \bar{\boldsymbol{o}}^{(\bar{k})}, (\bar{T}^{(\bar{k})}\boldsymbol{S}^{(k)})^{-1}) \\ &\quad \times \mathcal{W}(\boldsymbol{S}^{(k)} \mid \bar{T}^{(\bar{k})} + D, (\bar{T}^{(\bar{k})}\bar{\boldsymbol{C}}^{(\bar{k})})) \end{aligned} \tag{3.49}$$

where $\bar{T}^{(\bar{k})}$, $\bar{\boldsymbol{o}}^{(\bar{k})}$ and $\bar{\boldsymbol{C}}^{(\bar{k})}$ respectively denote the occupancy probability, the mean vector and the covariance matrix of a subset of training data $\boldsymbol{O}^{(\bar{k})}$. These parameters are efficiently computed in context clustering because it is assumed that the state occupancies

are not changed by splitting nodes. Moreover, the posterior distributions $Q(\boldsymbol{\mu}^{(k)}, \boldsymbol{S}^{(k)})$ can be estimated by Eqs. (3.23)–(3.28). Here, since the assumption the state occupancies are not changed by splitting nodes are used, the posterior distributions of all folds are represented by the same parameters. Therefore, although the Bayesian approach using cross validation increases the computational cost, the prior and posterior distributions are efficiently calculated in context clustering.

3.3 Experiments

To evaluate the effectiveness of the proposed method, speaker independent continuous phoneme recognition experiments were performed.

3.3.1 Experimental conditions

The 20,000 and 1,000 Japanese sentences uttered by male speakers from Japanese Newspaper Article Sentences (JNAS) [50] were used for model training. The 100 Japanese sentences uttered by male speakers, which were not included in the training data, from JNAS were used for evaluation. The average lengths of the training 20,000 utterances, training 1,000 utterances and test 100 utterances were 6.16 seconds, 6.42 seconds, and 5.83 seconds, respectively. Speech signals were sampled at a rate of 16 kHz and widowed at a 10 ms frame rate using a 25 ms Hamming window. The feature vectors consisted of the 0th through 13th mel-frequency cepstral coefficients (MFCCs), their delta and delta-delta coefficients. A three-state, left-to-right and no skip structure HMMs were used as triphone HMMs, and 204 questions were prepared in decision tree context clustering. In these experiments, we used a phoneme network imposing the constraints of Japanese phoneme transitions. However, phoneme N-gram probabilities and the language model weight were not used. The insertion penalty was adjusted for each experiment so that the number of insertion and deletion errors become almost equal. The experimental conditions are summarized in Table 3.1.

In recent HMM-based speech recognition systems, a multi-mixture Gaussian is typically used as a state output probability distribution. Although the VB method has been applied to multi-mixture HMMs [12], to evaluate the effect of only the proposed context clustering algorithm, each state output probability distribution was assumed to be modeled by a single Gaussian distribution with a diagonal covariance matrix in these experiments. Then, since the likelihood of each dimension is computed independently, the Gauss-Wishart distribution is equal to the Gauss-Gamma distribution.

Table 3.1: Experimental conditions.

Training data	JNAS 20,000 utterances JNAS 1,000 utterances
Test data	JNAS 100 utterances
Sampling rate	16 kHz
Feature vector	13-order MFCC + ΔMFCC + $\Delta\Delta$MFCC
Window	Hamming
Frame size	25ms
Frame shift	10ms
Number of HMM states	3 (left-to-right)
Number of phoneme categories	43

Table 3.2: K-fold cross validation (20,000 utterances).

	K				
	5	10	20	100	200
Number of states	14,072	14,360	14,474	14,575	14,610
Phoneme accuracy (%)	80.4	80.3	80.3	80.3	80.4

3.3.2 Number of folds in cross validation

In these experiments, the several number of folds in Bayesian context clustering using cross validation were compared. Table 3.2 and 3.3 show the number of states and phoneme accuracies with the acoustic models trained by 20,000 and 1,000 utterances, respectively, when the number of folds for cross validation were varied. As the number of folds increased, the computational cost was also proportionally increased and the resultant model structure became stable. Results show that the phoneme accuracy did not improve much with acoustic models trained by 20,000 utterances when the number of K was changed. However, in 1,000 utterances training condition, the phoneme accuracies were not stable. So, the large number of folds are required when the training data is small.

3.3.3 Comparison of conventional approaches

In these experiments, the following three approaches were compared.

- **ML-MDL** : Acoustic models were trained by the ML criterion and model structures were selected by the MDL criterion.

Table 3.3: K-fold cross validation (1,000 utterances).

	K				
	5	10	20	100	200
Number of states	3,919	4,065	4,101	4,141	4,156
Phoneme accuracy (%)	78.7	78.7	79.4	78.9	79.0

- **ML-CVML** : Acoustic models were trained by the ML criterion and model structures were selected by cross validation with the ML criterion.

- **Bayes-CVBayes** : Acoustic models were trained by the Bayesian criterion and model structures were selected by cross validation with the Bayesian criterion.

Figure 3.3 and 3.4 show the phoneme accuracies of acoustic models trained by 20,000 and 1,000 utterances, respectively. For **ML-CVML** and **Bayes-CVBayes**, 200-fold cross validation was used. To evaluate the performance of model selection, the phoneme accuracies with varying the size of decision trees are also shown. The decision trees were generated by changing a threshold of the stopping criterion $\Delta\mathcal{F} \leq threshold$ in the context clustering. In these figures, the lines represent the phoneme accuracies for each model structure and the points represent the phoneme accuracies of the model structure selected automatically by each method. These figures show that the proposed method **Bayes-CVBayes** selected the largest model structure, and the conventional method **ML-MDL** selected the smallest model structure in both training conditions. The model structure selected by **Bayes-CVBayes** was closer to that performed the highest accuracy than **ML-MDL**. Consequently, the proposed method **Bayes-CVBayes** outperforms the conventional method,**ML-MDL** and **ML-CVML**. In Fig. 3.3, **Bayes-CVBayes** achieved a 8.08% relative error reductions over **ML-MDL**.

It can be considered that the improvement of the proposed method caused by two factors, marginalization of model parameters and model selection. To discuss the impact of these two factors, an additional experiment was performed by swapping the model structures of **ML-MDL** and **Bayes-CVBayes**. The following two approaches were compared to **ML-MDL** and **Bayes-CVBayes**.

- **Bayes-MDL** : Acoustic models were trained by the Bayesian criterion and model structures selected by **ML-MDL** were used.

- **ML-CVBayes** : Acoustic models were trained by ML criterion and model structures selected by **Bayes-CVBayes** were used.

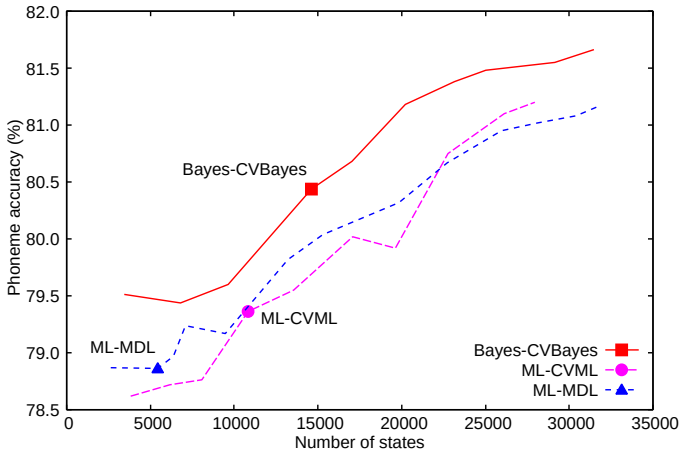

Figure 3.3: Phoneme accuracies of **ML-MDL**, **ML-CVML** and **Bayes-CVBayes** trained by 20,000 utterances versus the number of states.

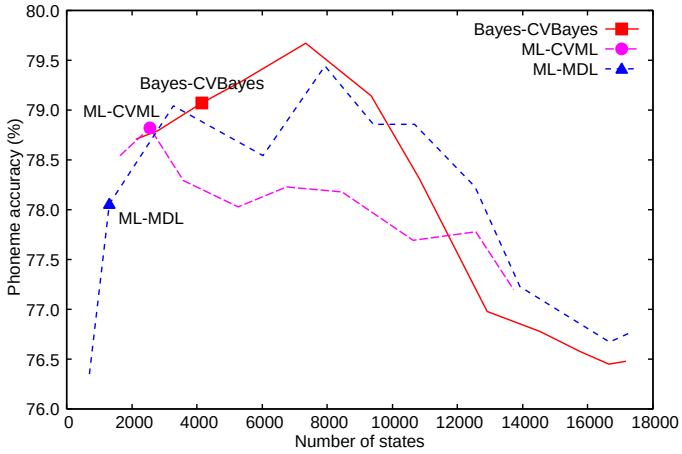

Figure 3.4: Phoneme accuracies of **ML-MDL**, **ML-CVML** and **Bayes-CVBayes** trained by 1,000 utterances versus the number of states.

Figure 3.5 and 3.6 show the phoneme accuracies of acoustic models trained by 20,000 and 1,000 utterances, respectively. Although the difference between **Bayes-MDL** and **ML-MDL** was the marginalization by the Bayesian approach, the phoneme accuracies of **Bayes-MDL** were improved from **ML-MDL** on both training conditions. Furthermore, the phoneme accuracies of **ML-CVBayes** were also improved when compared with **ML-MDL** on both training conditions, due to the model selection based on the Bayesian criterion with cross validations. Therefore, these results clearly showed that the Bayesian approach was effective for both the model training and the model selection. However, **Bayes-MDL** and **ML-CVBayes** were worse than **Bayes-CVBayes**. This means that train-

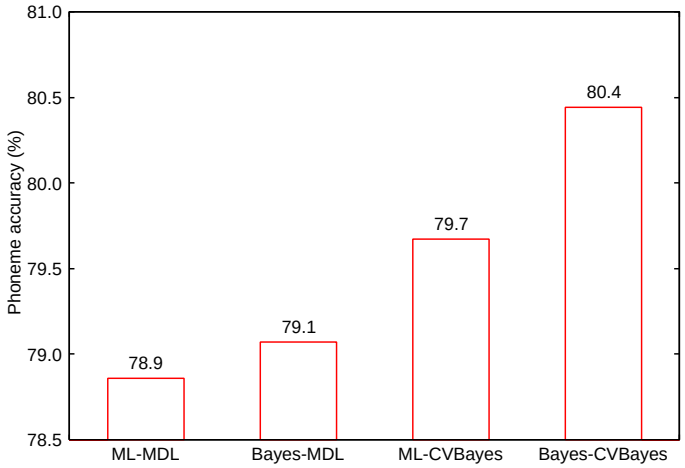

Figure 3.5: Phoneme accuracies when the acoustic models were trained by 20,000 utterances with the swapped decision tree.

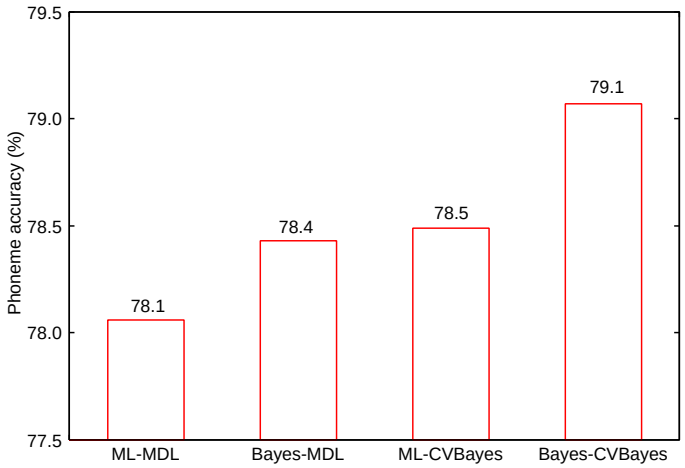

Figure 3.6: Phoneme accuracies when the acoustic models were trained by 1,000 utterances with the swapped decision tree.

ing criterion and model selection were strongly related, and these should be consistently performed based on the Bayesian criterion.

3.3.4 Marginal likelihood of the training and test data

Figure 3.7 and 3.8 show the relation among the lower bound $\mathcal{F}^{(\mathrm{CV})}$ for the training data, $\mathcal{F}$ for the test data with the correct phoneme sequences, and the phoneme accuracies. In these figures, a similar tendency between $\mathcal{F}^{(\mathrm{CV})}$ and $\mathcal{F}$ was observed, and the model struc-

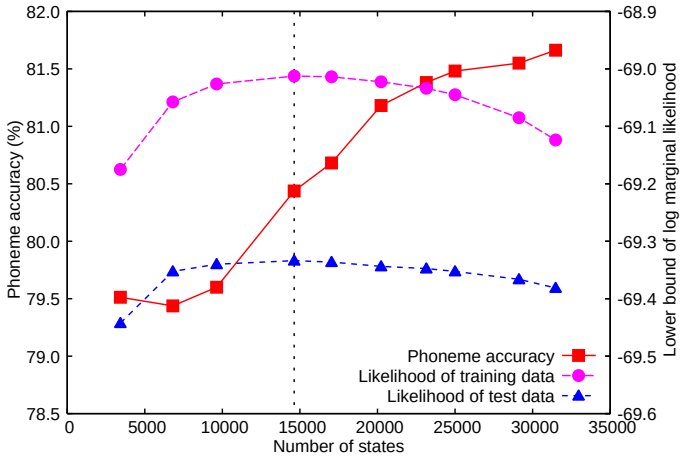

Figure 3.7: Log marginal likelihoods on both training and test data versus the number of states when the acoustic models were trained by 20,000 utterances.

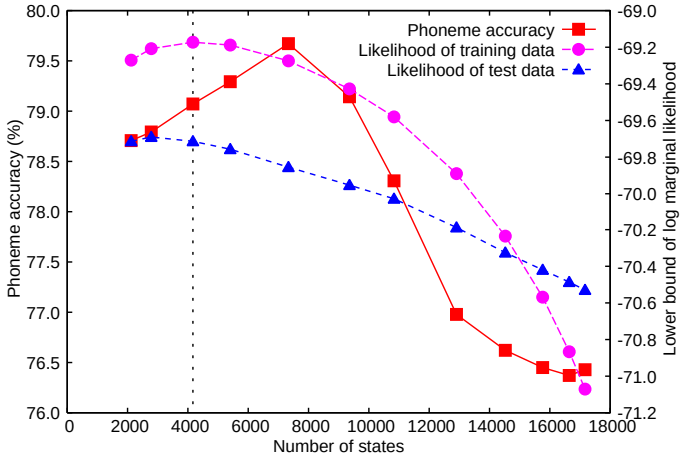

Figure 3.8: Log marginal likelihoods on both training and test data versus the number of states when the acoustic models were trained by 1,000 utterances.

ture which gave the highest $\mathcal{F}^{(\mathrm{CV})}$ also achieved the highest $\mathcal{F}$. However, the phoneme accuracy was not proportional to $\mathcal{F}$, and the proposed method could not select the model structure which achieved the highest phoneme accuracy. This means that although the proposed method could select the model structure which can accurately predict acoustic features for each HMM state, it is not identical to the performance in the classification problem. This is because the likelihood of incorrect phoneme sequences including insertion and deletion errors were not considered in the proposed method. This result suggests that a Bayesian criterion which can represent the classification performance directly is required.

3.4 Summary

This chapter proposed the Bayesian context clustering using cross validation for speech recognition based on the variational Bayesian framework. In the proposed method, the prior distributions are determined by using cross validation, and the determined prior distribution is applied to the context clustering. The results on continuous phoneme recognition experiments demonstrated that the proposed method outperformed the context clustering based on the MDL criterion and cross validation with ML estimates. The proposed method could determine prior distributions without any tuning parameters, and select the model structure which can accurately predict acoustic features for each HMM state. As future work, we will apply a Bayesian criterion using cross validation for selecting the number of mixtures, and apply a Bayesian criterion which represents the classification performance directly to the context clustering.

Chapter 4

Bayesian Speech Synthesis

A statistical speech synthesis system based on hidden Markov models (HMMs) was recently developed. In HMM-based speech synthesis, the spectrum, excitation and duration of speech are modeled simultaneously with HMMs, and speech parameter sequences are generated from the HMMs themselves [16].

In HMM-based speech synthesis, the maximum likelihood (ML) criterion has been typically used for training HMMs and generating speech parameters. The ML criterion guarantee that the ML estimates approach the true values of the parameters. Therefore, acoustic modeling based on HMMs has been developed greatly by using the ML approach. However, since the ML criterion produces a point estimate of the HMM parameters, its estimation accuracy may deteriorate when the amount of training data is insufficient.

The Bayesian approach considers the posterior distribution of any variable, as well as the prior distribution [7]. That is, all the variables introduced when the models are parameterized, such as the model parameters and latent variables, are regarded as probabilistic variables, and their posterior distributions are obtained by invoking Bayes theorem. Moreover, the Bayesian approach can utilize prior information. The prior information of the model parameters is represented by prior distributions, as well as posterior distributions. The difference between the Bayesian and ML approaches is that the target of estimation is the distribution function in the Bayesian approach whereas it is the parameter value in the ML approach. Based on the posterior distribution estimation, the Bayesian approach can generally construct a more robust model than the ML approach. However, the Bayesian approach requires complex integral and expectation computations to obtain posterior distributions when the models have latent variables. To overcome this problem, the maximum a posterior (MAP) criterion has been proposed [48]. The MAP criterion can utilize prior information, but cannot obtain the posterior distributions. Therefore, the MAP criterion leads th the lack of robustness. Recently, a Variational Bayes method [17]

has been proposed in the learning theory field. This method can obtain approximate posterior distributions through iterative calculations similar to the expectation-maximization (EM) algorithm used in the ML approach. The variational Bayes method has been applied to speech recognition and it shows good performance [8–11].

In a real speech, there are a number of contextual factors that affect spectrum, excitation and duration of speech (e.g., phone identity, accent, stress). By considering the context, more accurate acoustic models are estimated. Therefore, context-dependent models are typically used to capture these factors in HMM-based speech synthesis [42]. Although a large number of context-dependent models can capture variations in speech data, too many model parameters lead to the over-fitting problem. Consequently, maintaining a proper balance between model complexity and the amount of training data is required. The decision tree based context clustering [43] is a successful method for context-dependent HMM estimation to deal with the problem of training data insufficiency, not only for the robust parameter estimation but also for predicting probability distributions for unseen contexts. This method constructs a parameter tying structure which can assign a sufficient amount of training data to each HMM state. A binary tree is grown step by step, by choosing a question which divides the context using a greedy strategy to maximize some objective function. The ML criterion is inappropriate as a model selection criterion since the it increases monotonically as the number of states increases. Some heuristic thresholding is therefore necessary to stop splitting nodes in context clustering. To solve this problem, the minimum description length (MDL) criterion has been employed to select the model structure in the speech synthesis field [44]. The MDL criterion is performed as the ML criterion with the penalty term, and the penalty term can be derived automatically. However, since the MDL criterion is based on an asymptotic assumption, it is ineffective when the amount of training data is small.

On the other hand, since the Bayesian approach does not use an asymptotic assumption, unlike the MDL criterion, it is available even in the case of insufficient amounts of training data. In the Bayesian approach, an appropriate model structure can be selected by maximizing the marginal likelihood. Bayesian information criterion (BIC) [45] have been proposed as an approximated Bayesian criterion, but BIC and MDL are practically the same. Consequently, BIC is ineffective when the amount of training data is small. The VB method is an attractive alternative to BIC for model selection problem. The VB method can select appropriate model structure, even when there are insufficient amounts of data, because it does not use an asymptotic assumption, unlike the BIC or MDL In the VB method, an appropriate model structure can be selected by maximizing the marginal likelihood [11, 12].

This chapter proposes a new framework of speech synthesis based on the Bayesian ap-

proach. In this framework, all processes for constructing the system are derived from one single predictive distribution which exactly represents the problem of speech synthesis. The Bayesian approach assumes that model parameters are random variables and reliable predictive distributions are estimated by marginalizing model parameters. In this chapter, the VB method is consistently employed to estimate posterior distributions of latent variables and select model structures.

The rest of this chapter is organized as follows. Section 4.1 describes the Bayesian approach to speech synthesis. Section 4.2 describes HSMM based Bayesian speech synthesis. In Section 4.3, subjective listening test results are presented. Furthermore, the Bayesian speech synthesis framework integrating the training and synthesis processes is proposed in Section 4.4. Section 4.5 shows subjective listening test results of the integration method. Concluding remarks and future work are presented in final section.

4.1 Bayesian Speech synthesis

4.1.1 Bayesian approach

In HMM-based speech synthesis, the ML criterion has been typically used to train HMMs and generate speech parameters. The optimal model parameters can be obtained by maximizing the likelihood for a given training data as follows:

$$\boldsymbol{\Lambda}_{ML} = \arg\max_{\boldsymbol{\Lambda}} P(\boldsymbol{O} \mid S, \boldsymbol{\Lambda}) \,, \tag{4.1}$$

where S is a label sequence of training data. Since it is difficult to analytically obtain the model parameter $\boldsymbol{\Lambda}_{ML}$, the model parameter can be obtained using an iterative procedure such as the expectation-maximization (EM) algorithm. In the synthesis part, the speech parameter generation algorithm generates sequences of speech parameter vectors that maximize their output probabilities by using the model parameters $\boldsymbol{\Lambda}_{ML}$.

$$\boldsymbol{o}_{ML} = \arg\max_{\boldsymbol{o}} P(\boldsymbol{o} \mid s, \boldsymbol{\Lambda}_{ML}) \,, \tag{4.2}$$

where $\boldsymbol{o} = \left[\boldsymbol{o}_1^\top, \boldsymbol{o}_2^\top, \ldots, \boldsymbol{o}_T^\top\right]^\top$ is a speech parameter sequence and s is a label sequence to be synthesized. However, since the ML estimator produces a point estimate of the HMM parameters, the estimation accuracy may deteriorate when the amount of training data is insufficient.

The Bayesian approach assumes that a set of model parameters $\boldsymbol{\Lambda}$ is a random variable, while the ML approach estimates constant model parameters. In the Bayesian approach,

the speech parameter is generated from a predictive distribution as follows.

$$\begin{aligned} \boldsymbol{o}_{Bayes} &= \arg\max_{\boldsymbol{o}} P(\boldsymbol{o} \mid s, \boldsymbol{O}, S) \\ &= \arg\max_{\boldsymbol{o}} P(\boldsymbol{o}, \boldsymbol{O} \mid s, S) \end{aligned} \tag{4.3}$$

It can be seen that Eq. (4.3) directly represents the problem of speech synthesis; that is, the speech feature sequence $\boldsymbol{o}$ is generated from given training feature sequences $\boldsymbol{O}$ with labels S and labels to be synthesized s. The marginal likelihood of $\boldsymbol{o}$ and $\boldsymbol{O}$ is defined by

$$\begin{aligned} P(\boldsymbol{o}, \boldsymbol{O} \mid s, S) &= \sum_{\boldsymbol{z}} \sum_{\boldsymbol{Z}} \int P(\boldsymbol{o}, \boldsymbol{z}, \boldsymbol{O}, \boldsymbol{Z}, \boldsymbol{\Lambda} \mid s, S) d\boldsymbol{\Lambda} \\ &= \sum_{\boldsymbol{z}} \sum_{\boldsymbol{Z}} \int P(\boldsymbol{o}, \boldsymbol{z} \mid s, \boldsymbol{\Lambda}) P(\boldsymbol{O}, \boldsymbol{Z} \mid S, \boldsymbol{\Lambda}) P(\boldsymbol{\Lambda}) d\boldsymbol{\Lambda} \end{aligned} \tag{4.4}$$

where $\boldsymbol{z}$ is a sequence of HMM states for a speech parameter sequence $\boldsymbol{o}$, $P(\boldsymbol{\Lambda})$ is the prior distribution for model parameter $\boldsymbol{\Lambda}$, $P(\boldsymbol{o}, \boldsymbol{z} \mid s, \boldsymbol{\Lambda})$ is the likelihood of synthesis data $\boldsymbol{o}$, and $P(\boldsymbol{O}, \boldsymbol{Z} \mid S, \boldsymbol{\Lambda})$ is the likelihood of training data $\boldsymbol{O}$. The model parameters are integrated out in Eq. (4.4) so that the effect of over-fitting is mitigated. However, it is difficult to solve the integral and expectation calculations. The calculations become more complicated when a model includes latent variables. The variational Bayes method has been proposed as a tractable approximation method to overcome this problem, and it has good generalization performance in many applications [17].

4.1.2 Variational Bayes method for speech synthesis

The variational Bayes method maximizes the lower bound of the log marginal likelihood $\mathcal{F}$ instead of the true marginal likelihood. The lower bound $\mathcal{F}$ is defined by using Jensen's inequality.

$$\begin{aligned} \log P(\boldsymbol{o}, \boldsymbol{O} \mid s, S) &= \log \sum_{\boldsymbol{z}} \sum_{\boldsymbol{Z}} \int P(\boldsymbol{o}, \boldsymbol{z}, \boldsymbol{O}, \boldsymbol{Z}, \boldsymbol{\Lambda} \mid s, S) d\boldsymbol{\Lambda} \\ &= \log \sum_{\boldsymbol{z}} \sum_{\boldsymbol{Z}} \int Q(\boldsymbol{z}, \boldsymbol{Z}, \boldsymbol{\Lambda}) \frac{P(\boldsymbol{o}, \boldsymbol{z}, \boldsymbol{O}, \boldsymbol{Z}, \boldsymbol{\Lambda} \mid s, S)}{Q(\boldsymbol{z}, \boldsymbol{Z}, \boldsymbol{\Lambda})} d\boldsymbol{\Lambda} \\ &\geq \sum_{\boldsymbol{z}} \sum_{\boldsymbol{Z}} \int Q(\boldsymbol{z}, \boldsymbol{Z}, \boldsymbol{\Lambda}) \log \frac{P(\boldsymbol{o}, \boldsymbol{z}, \boldsymbol{O}, \boldsymbol{Z}, \boldsymbol{\Lambda} \mid s, S)}{Q(\boldsymbol{z}, \boldsymbol{Z}, \boldsymbol{\Lambda})} d\boldsymbol{\Lambda} \\ &= \left\langle \log \frac{P(\boldsymbol{o}, \boldsymbol{z}, \boldsymbol{O}, \boldsymbol{Z}, \boldsymbol{\Lambda} \mid s, S)}{Q(\boldsymbol{z}, \boldsymbol{Z}, \boldsymbol{\Lambda})} \right\rangle_{Q(\boldsymbol{z}, \boldsymbol{Z}, \boldsymbol{\Lambda})} \\ &= \mathcal{F} \end{aligned} \tag{4.5}$$

where $\langle\cdot\rangle_Q$ denotes a calculation of the expectation with respect to Q, and $Q(\boldsymbol{z}, \boldsymbol{Z}, \boldsymbol{\Lambda})$ is an approximate distribution of the true posterior distribution $P(\boldsymbol{z}, \boldsymbol{Z}, \boldsymbol{\Lambda} \mid \boldsymbol{o}, \boldsymbol{O}, s, S)$. The variational Bayes method uses the assumption that probabilistic variables associated with $\boldsymbol{z}, \boldsymbol{Z}, \boldsymbol{\Lambda}$ are statistically independent of the other variables.

$$Q(\boldsymbol{z}, \boldsymbol{Z}, \boldsymbol{\Lambda}) = Q(\boldsymbol{z})\, Q(\boldsymbol{Z})\, Q(\boldsymbol{\Lambda}) \tag{4.6}$$

The variational Bayes method uses the posterior distributions $Q(\boldsymbol{z})$, $Q(\boldsymbol{Z})$ and $Q(\boldsymbol{\Lambda})$ to approximate the true posterior distributions. The optimal posterior distributions can be obtained by maximizing the objective function $\mathcal{F}$ with the variational method as follows.

$$Q(\boldsymbol{z}) = C_{\boldsymbol{z}} \exp \langle \log P(\boldsymbol{o}, \boldsymbol{z} \mid s, \boldsymbol{\Lambda}) \rangle_{Q(\boldsymbol{\Lambda})} \tag{4.7}$$

$$Q(\boldsymbol{Z}) = C_{\boldsymbol{Z}} \exp \langle \log P(\boldsymbol{O}, \boldsymbol{Z} \mid S, \boldsymbol{\Lambda}) \rangle_{Q(\boldsymbol{\Lambda})} \tag{4.8}$$

$$\begin{aligned} Q(\boldsymbol{\Lambda}) = C_{\boldsymbol{\Lambda}} P(\boldsymbol{\Lambda}) &\exp \langle \log P(\boldsymbol{o}, \boldsymbol{z} \mid s, \boldsymbol{\Lambda}) \rangle_{Q(\boldsymbol{z})} \\ &\times \exp \langle \log P(\boldsymbol{O}, \boldsymbol{Z} \mid S, \boldsymbol{\Lambda}) \rangle_{Q(\boldsymbol{Z})} \end{aligned} \tag{4.9}$$

where $C_{\boldsymbol{z}}$, $C_{\boldsymbol{Z}}$ and $C_{\boldsymbol{\Lambda}}$ are the normalization terms of $Q(\boldsymbol{z})$, $Q(\boldsymbol{Z})$ and $Q(\boldsymbol{\Lambda})$, respectively. However, in the above algorithm, the optimal posterior distributions depend on synthesized speech parameter $\boldsymbol{o}$, i.e., the posterior distributions given a label sequence of synthesis speech are estimated. In a basic speech synthesis situation, the observed data for the synthesis sentences is not given beforehand. Therefore, the posterior distributions cannot be obtained. To avoid this problem, this chapter assumes that $Q(\boldsymbol{\Lambda})$ is independent of speech parameter $\boldsymbol{o}$. Then, $Q(\boldsymbol{\Lambda})$ is given by

$$Q(\boldsymbol{\Lambda}) = C_{\boldsymbol{Z}} P(\boldsymbol{\Lambda}) \exp \langle \log P(\boldsymbol{O}, \boldsymbol{Z} \mid S, \boldsymbol{\Lambda}) \rangle_{Q(\boldsymbol{Z})} \ . \tag{4.10}$$

By the above approximation, the posterior distribution $Q(\boldsymbol{\Lambda})$ can be estimated by only training data.

It is assumed that the model parameters $\boldsymbol{\pi} = \{\pi_i\}_{i=1}^N$, $\boldsymbol{a}_i = \{a_{ij}\}_{j=1}^N$, and $\{\boldsymbol{\mu}_i, \boldsymbol{S}_i\}_{i=1}^N$ are independent each other in the prior distribution $P(\boldsymbol{\Lambda})$. Therefore, the prior distribution $P(\boldsymbol{\Lambda})$ can be represented as follows.

$$P(\boldsymbol{\Lambda}) = P(\boldsymbol{\pi}) \prod_{i=1}^{N} P(\boldsymbol{a}_i) \prod_{i=1}^{N} P(\boldsymbol{\mu}_i, \boldsymbol{S}_i) \tag{4.11}$$

By using this assumption, the posterior distribution $Q(\boldsymbol{\Lambda})$ and its normalization term $C_{\boldsymbol{\Lambda}}$ can be written as follows.

$$Q(\boldsymbol{\Lambda}) = Q(\boldsymbol{\pi}) \prod_{i=1}^{N} Q(\boldsymbol{a}_i) \prod_{i=1}^{N} Q(\boldsymbol{\mu}_i, \boldsymbol{S}_i) \tag{4.12}$$

$$C_{\boldsymbol{\Lambda}} = C_{\boldsymbol{\pi}} \prod_{i=1}^{N} C_{\boldsymbol{a}_i} \prod_{i=1}^{N} C_{\boldsymbol{\mu}_i, \boldsymbol{S}_i} \tag{4.13}$$

From Eqs. (3.1), (3.2) and (4.9)–(4.13), the posterior distributions of model parameters are given as follows.

$$Q(\boldsymbol{\pi}) = C_{\boldsymbol{\pi}} P(\boldsymbol{\pi}) \exp\left\{\sum_{i=1}^{N} \langle Z_1^i \rangle \log \pi_i\right\} \tag{4.14}$$

$$Q(\boldsymbol{a}_i) = C_{\boldsymbol{a}_i} P(\boldsymbol{a}_i) \exp\left\{\sum_{i=1}^{N}\sum_{t=1}^{T-1} \langle Z_t^i Z_{t+1}^j \rangle \log a_{ij}\right\} \tag{4.15}$$

$$\begin{aligned} Q(\boldsymbol{\mu}_i, \boldsymbol{S}_i) = {} & C_{\boldsymbol{\mu}_i, \boldsymbol{S}_i} P(\boldsymbol{\mu}_i, \boldsymbol{S}_i) \\ & \times \exp\left\{\sum_{t=1}^{T} \langle Z_t^i \rangle \log \mathcal{N}(\boldsymbol{o}_t \mid \boldsymbol{\mu}_i, \boldsymbol{S}_i)\right\} \end{aligned} \tag{4.16}$$

where $\langle Z_t^i \rangle$ and $\langle Z_t^i Z_{t+1}^j \rangle$ are the expectation value with respect to $Q(\boldsymbol{Z})$ as follows.

$$\langle Z_t^i \rangle = \sum_{\boldsymbol{Z}} Q(\boldsymbol{Z}) Z_t^i \tag{4.17}$$

$$\langle Z_t^i Z_{t+1}^j \rangle = \sum_{\boldsymbol{Z}} Q(\boldsymbol{Z}) Z_t^i Z_{t+1}^j \tag{4.18}$$

The posterior distribution $Q(\boldsymbol{Z})$ are represented as follows.

$$\begin{aligned} Q(\boldsymbol{Z}) = {} & C_{\boldsymbol{Z}} \prod_{i=1}^{N} \exp\left\{Z_1^i \left\langle \log \pi_i \right\rangle_{Q(\boldsymbol{\pi})}\right\} \\ & \times \prod_{t=1}^{\hat{T}-1}\prod_{i=1}^{N} \exp\left\{Z_t^i Z_{t+1}^j \left\langle \log a_{ij} \right\rangle_{Q(\boldsymbol{a}_i)}\right\} \\ & \times \prod_{t=1}^{\hat{T}}\prod_{i=1}^{N} \exp\left\{Z_t^i \left\langle \log \mathcal{N}(\boldsymbol{o}_t \mid \boldsymbol{\mu}_i, \boldsymbol{S}_i^{-1}) \right\rangle_{Q(\boldsymbol{\mu}_i, \boldsymbol{S}_i)}\right\}. \end{aligned} \tag{4.19}$$

where $Q(\boldsymbol{Z})$ is same with the likelihood function of HMM, and Eq. (4.19) can be computed efficiently by the forward-backward algorithm. These optimizations can be effectively performed by iterative calculations as the EM algorithm, which increases the value of objective function $\mathcal{F}$ at each iteration until convergence.

4.1.3 Speech parameter generation

In HMM-based speech synthesis, rhythm and tempo are controlled by state duration probability distributions. One of major limitation of HMMs is that they do not provide an adequate representation of the temporal structure of speech. This is because the probability of state occupancy decreases exponentially with time. To overcome this limitation,

in HMM-based speech synthesis system, each state duration probability distribution is explicitly modeled by a single Gaussian distribution. They are estimated from statistics obtained in the last iteration of the forward-backward algorithm, and then clustered by the decision tree-based context clustering [42, 43]. In the synthesis part, we construct a sentence HMM corresponding to an arbitrarily given text and determine state durations which maximize their probabilities. Then, a speech parameter sequence is generated for the given state sequence by the speech parameter generation algorithm [51].

In the synthesis part, first an arbitrarily given text to be synthesized is converted to a context-dependent label sequence and a sentence HMM is constructed by concatenating context-dependent HMMs according to the label sequence. Secondly, state durations $\boldsymbol{d}$ of the sentence HMM $\boldsymbol{\Lambda}$ are determined as follows.

$$\boldsymbol{d}_{max} = \arg\max_{\boldsymbol{d}} \langle \log P(\boldsymbol{d} \mid \boldsymbol{\Lambda}) \rangle_{Q(\boldsymbol{\Lambda})} \tag{4.20}$$

Thirdly, a speech parameter sequence is generated for a given state sequence. We assume that a speech parameter vector $\boldsymbol{o}_t$ consists of a static feature vector $\boldsymbol{c}_t$ and its first and second order dynamic feature vectors, that is

$$\begin{aligned} \boldsymbol{o} &= \boldsymbol{W}\boldsymbol{c} \\ &= \left[(\boldsymbol{W}\boldsymbol{c})_1^\top, (\boldsymbol{W}\boldsymbol{c})_2^\top, \ldots, (\boldsymbol{W}\boldsymbol{c})_T^\top \right]^\top \qquad (4.21) \\ (\boldsymbol{W}\boldsymbol{c})_t &= \left[\boldsymbol{c}_t^\top, \Delta \boldsymbol{c}_t^\top, \Delta^2 \boldsymbol{c}_t^\top \right]^\top \qquad (4.22) \end{aligned}$$

where $\boldsymbol{W}$ is a window matrix to calculate dynamic features from static features [51]. In the synthesis part, a static feature vector sequence $\boldsymbol{c}$ is generated. By the variational Bayes method, the lower bound $\mathcal{F}$ approximates the log marginal likelihood $\log P(\boldsymbol{W}\boldsymbol{c}, \boldsymbol{O} \mid s, S)$. Therefore, the optimal speech parameter sequence $\hat{\boldsymbol{c}}$ is generated by maximizing the lower bound $\mathcal{F}$:

$$\begin{aligned} \frac{\partial \mathcal{F}}{\partial \boldsymbol{c}} &= \frac{\partial}{\partial \boldsymbol{c}} \langle \log P(\boldsymbol{W}\boldsymbol{c} \mid \boldsymbol{z}, \boldsymbol{\Lambda}) P(\boldsymbol{z} \mid s, \boldsymbol{\Lambda}) \rangle_{Q(\boldsymbol{z})Q(\boldsymbol{\Lambda})} \\ &= \left\langle \frac{\partial}{\partial \boldsymbol{c}} \log P(\boldsymbol{W}\boldsymbol{c} \mid \boldsymbol{z}, \boldsymbol{\Lambda}) \right\rangle_{Q(\boldsymbol{\Lambda})} = \boldsymbol{0} \,. \end{aligned} \tag{4.23}$$

Under the condition in Eq. (4.21), the optimal static feature sequence $\hat{\boldsymbol{c}}$ can be determined by solving the following set of linear equations:

$$\boldsymbol{W}^\top \langle \boldsymbol{S} \rangle \boldsymbol{W} \hat{\boldsymbol{c}} = \boldsymbol{W}^\top \langle \boldsymbol{S}\boldsymbol{\mu} \rangle \,, \tag{4.24}$$

where $\langle \boldsymbol{S} \rangle$ and $\langle \boldsymbol{S}\boldsymbol{\mu} \rangle$ represent the expectation value of $\boldsymbol{S}$ and $\boldsymbol{S}\boldsymbol{\mu}$ with respect to $Q(\boldsymbol{\Lambda})$, respectively. In the speech generation based on the ML criterion, the optimal static feature sequence can be determined by solving the following equation:

$$\boldsymbol{W}^\top \boldsymbol{S} \boldsymbol{W} \hat{\boldsymbol{c}} = \boldsymbol{W}^\top \boldsymbol{S}\boldsymbol{\mu} \,, \tag{4.25}$$

Eq. (4.25) can be solved efficiently using the Cholesky or QR decomposition [51]. By the same token, Eq. (4.24) can be solved and the computational cost is almost the same as the ML criterion.

4.2 HSMM based Bayesian speech synthesis

In the HMM-based speech synthesis system, each state duration probability distribution is explicitly modeled by a single Gaussian distribution. They are estimated from statistics obtained in the last iteration of the forward-backward algorithm, and then clustered by the decision tree-based context clustering [42, 43]. However, there is an inconsistency between training and synthesis: although speech is synthesized from HMMs with explicit state duration probability distributions, HMMs are trained without them. To overcome this inconsistency, hidden semi-Markov model (HSMM) based speech synthesis has been proposed [52]. This framework introduces an HSMM, which is an HMM with explicit state duration probability distributions, into not only for synthesis but also training in the HMM-based speech synthesis system.

4.2.1 Likelihood computation of the HMM

The model likelihood of an HMM $\mathbf{\Lambda}$ for an observation vector sequence $\boldsymbol{O} = (\boldsymbol{O}_1, \ldots, \boldsymbol{O}_T)$ can be computed efficiently by the forward-backward algorithm. First, we define partial forward likelihood $\alpha_t(\cdot)$ as follows.

$$\begin{aligned}\alpha_t(j) =& P(\boldsymbol{O}_1, \ldots, \boldsymbol{O}_t, Z_t = j \mid \mathbf{\Lambda}) \\ =& \sum_{i=1}^{N} \alpha_{t-1}(i) a_{ij} b_j(\boldsymbol{O}_t), \\ & (1 \leq t \leq T, 1 \leq j \leq N)\end{aligned} \tag{4.26}$$

where a_{ij} is a state transition probability from i-th state to j-th state, $b_j(\boldsymbol{O}_t)$ is an output probability of observation vector $\boldsymbol{O}_t$ from j-th state, N is a total number of HMM states. To begin the recursion Eq. (4.26), we set $\alpha_1(j) = \pi_j b_j(\boldsymbol{O}_1), 1 \leq j \leq N$, where π_j is an initial state probability of j-th state. Secondly, partial backward likelihood $\beta_t(\cdot)$ is defined

as follows.

$$\begin{aligned}\beta_t(i) =& P(\boldsymbol{O}_{t+1}, \ldots, \boldsymbol{O}_T \mid Z_{t+1} = i, \boldsymbol{\Lambda}) \\ =& \sum_{j=1}^{N} a_{ij} b_j(\boldsymbol{O}_{t+1}) \beta_{t+1}(i), \\ & (1 \leq t \leq T, 1 \leq i \leq N)\end{aligned} \tag{4.27}$$

To begin the recursion Eq. (4.27), we set $\beta_T(i) = 1, 1 \leq i \leq N$. From Eqs. (4.26) and (4.27), the model likelihood $P(\boldsymbol{O} \mid \boldsymbol{\Lambda})$ is computed as

$$P(\boldsymbol{O} \mid \boldsymbol{\Lambda}) = \sum_{i=1}^{N} \alpha_t(i) \cdot \beta_t(i). \quad 1 \leq t \leq T \tag{4.28}$$

Generally, computational complexity of the above recursions is on the order of $O(N^2 T)$. However, if a simple left-to-right structure is assumed, it reduces to $O(NT)$.

4.2.2 Likelihood computation of the HSMM

The model likelihood of an HSMM $\boldsymbol{\Lambda}'$ for an observation vector sequence $\boldsymbol{O}$ can be computed efficiently by the generalized forward-backward algorithm. We can compute partial forward likelihood $\alpha'_t(\cdot)$ and partial backward likelihood $\beta'_t(\cdot)$ recursively as follows.

$$\alpha'_0(j) = \pi_j, \tag{4.29}$$

$$\begin{aligned}\alpha'_t(j) =& \sum_{d=1}^{t} \sum_{i=1, j \neq j}^{N'} \alpha'_{t-d}(i) a'_{ij} p'_j(d) \\ & \times \prod_{s=t-d+1}^{t} b'_j(\boldsymbol{O}_s), \quad 1 \leq t \leq T\end{aligned} \tag{4.30}$$

$$\beta'_T(i) = 1, \tag{4.31}$$

$$\begin{aligned}\beta'_t(i) =& \sum_{d=1}^{T-t} \sum_{j=1, j \neq i}^{N'} a'_{ij} p'_j(d) \\ & \times \prod_{s=t+1}^{t+d} b'_j(\boldsymbol{O}_s) \beta'_{t+d}(j), \quad 1 \leq t \leq T\end{aligned} \tag{4.32}$$

where $a'_{ij}, b'_j(\boldsymbol{O}_t), N' p'_j(d)$, and π'_j are a state transition probability from i-th state to j-th state, an output probability of observation vector $\boldsymbol{O}_t$ from j-th state, a total number of HSMM states, a state duration probability of j-th state, and an initial state probability of

j-th state, respectively. From above equations, the model likelihood $P(\boldsymbol{O} \mid \boldsymbol{\Lambda}')$ is given by

$$
\begin{aligned}
P(\boldsymbol{O} \mid \boldsymbol{\Lambda}') &= \sum_{i=1}^{N'} \sum_{j=1, i \neq j}^{N'} \sum_{d=1}^{t} \alpha'_{t-d}(i) a'_{ij} p'_j(d) \\
&\quad \times \prod_{s=t-d+1}^{t} b'_j(\boldsymbol{O}_s) \beta'_t(j).
\end{aligned} \tag{4.33}
$$

The drawback of the HSMMs is that the above recursions require on the order of $O(N'^2 T^2)$ calculations, as compared with $O(N'^2 T)$ of the HMM. If a simple left-to-right structure is assumed, it reduces to $O(N' T^2)$. Furthermore, by limiting the maximum duration to D, it further reduces to $O(N' D T)$. Although the use of HSMMs increases computational cost, it is still possible to perform the above recursions using the currently available computational resources.

4.2.3 Optimization of posterior distributions

In HSMM-based Bayesian speech synthesis, the optimizations using Eqs. (4.7), (4.8), and (4.9) can be effectively performed by iterative calculations as the expectation maximization (EM) algorithm, which increases the value of objective function $\mathcal{F}$ at each iteration until convergence. The normalization term $C_{\boldsymbol{Z}}$ of an HSMM can be computed efficiently by the generalized forward-backward algorithm for the variational Bayes method.

$$
\begin{aligned}
C_{\boldsymbol{Z}}^{-1} &= \sum_{\boldsymbol{Z}} \exp \left\langle \log P(\boldsymbol{O}, \boldsymbol{Z} \mid s, \boldsymbol{\Lambda}) \right\rangle_{Q(\boldsymbol{\Lambda})} \\
&= \sum_{i=1}^{\hat{N}} \sum_{j=1, i \neq j}^{\hat{N}} \sum_{d=1}^{t} \hat{\alpha}_{t-d}(i) \exp \langle \log a_{ij} \rangle_{Q(\boldsymbol{\Lambda})} \\
&\quad \times \exp \langle \log p_j(d) \rangle_{Q(\boldsymbol{\Lambda})} \\
&\quad \times \prod_{s=t-d+1}^{t} \exp \langle \log b_j(\boldsymbol{O}_s) \rangle_{Q(\boldsymbol{\Lambda})} \hat{\beta}_t(j).
\end{aligned} \tag{4.34}
$$

We can compute partial forward likelihood $\hat{\alpha}_t(\cdot)$ and partial backward likelihood $\hat{\beta}_t(\cdot)$ recursively as follows:

$$\begin{aligned}
\hat{\alpha}_t(j) &= \sum_{d=1}^{t} \sum_{i=1, i\neq j}^{\hat{N}} \hat{\alpha}_{t-d}(i) \exp\langle \log a_{ij} \rangle_{Q(\mathbf{\Lambda})} \\
&\quad \times \exp\langle \log p_j(d) \rangle_{Q(\mathbf{\Lambda})} \\
&\quad \times \prod_{s=t-d+1}^{t} \exp\langle \log b_j(\boldsymbol{O}_s) \rangle_{Q(\mathbf{\Lambda})}, \qquad (4.35) \\
\hat{\beta}_t(i) &= \sum_{d=1}^{T-t} \sum_{j=1, j\neq i}^{\hat{N}} \exp\langle \log a_{ij} \rangle_{Q(\mathbf{\Lambda})} \\
&\quad \times \exp\langle \log p_j(d) \rangle_{Q(\mathbf{\Lambda})} \\
&\quad \times \prod_{s=t+1}^{t+d} \exp\langle \log b_j(\boldsymbol{O}_s) \rangle_{Q(\mathbf{\Lambda})} \hat{\beta}_{t+d}(j). \qquad (4.36)
\end{aligned}$$

Because the Bayesian approach assumes that a set of model parameters $\mathbf{\Lambda}$ is a random variable, model parameters are represented by the expectation values. The normalization term $C_{\boldsymbol{Z}}$ can be computed as like Eq. (4.34). Although the computational cost is increased by using HSMMs, the Bayesian approach requires almost the same computational cost with the ML criterion.

In the Bayesian approach, a conjugate prior distribution is widely used as a prior distribution $P(\mathbf{\Lambda})$. When the state duration probability distribution is a Gaussian distribution, the conjugate prior distribution becomes a Gauss-Gamma distribution:

$$P(\boldsymbol{\mu}, \boldsymbol{S}) = \mathcal{N}(\boldsymbol{\mu} \mid \nu, (\xi\Sigma)^{-1}) \, \mathcal{G}\left(\Sigma \mid \frac{\eta}{2}, \frac{B}{2}\right) , \qquad (4.37)$$

where $\{\xi, \eta, \nu, B\}$ is a hyper-parameter set. Using a conjugate prior distribution, a set of parameters of posterior distribution is also represented by the same parameter set $\{\bar{\xi}, \bar{\eta}, \bar{\boldsymbol{\nu}}, \bar{\boldsymbol{B}}\}$.

4.3 Experiments

4.3.1 Experimental conditions

The experiments used the ATR Japanese speech database [53] B-set, which consists of 503 phonetically balanced sentences. The first 450 of the 503 sentences, uttered by one

male speaker (MHT), were used for training. The remaining 53 sentences were used for the evaluations. Speech signals were sampled at a rate of 16 kHz and windowed at a 5-ms frame rate using a 25-ms Blackman window. Feature vectors consisted of spectrum and F_0 parameter vectors. The spectrum parameter vectors consisted of 24 mel-cepstral coefficients, and their delta and delta-delta coefficients. The F_0 parameter vectors consisted of log F_0, and its delta and delta-delta. A five-state, left-to-right MSD-HMM [54] and MSD-HSMM without skip transitions was used. Each state output PDF was composed of spectrum and F_0 streams. The spectrum stream was modeled by single multi-variate Gaussian distributions with diagonal covariance matrices. The F_0 stream was modeled by a multi-space probability distribution consisting of a Gaussian distribution for voiced frames and a discrete distribution for unvoiced frames. Each state duration PDF was modeled by a one-dimensional Gaussian distribution. The decision tree-based context clustering technique was separately applied to distributions of spectrum, F_0, and state duration.

A subjective listening test was conducted to evaluate the quality of the synthesized speech. The test assessed the naturalness of the synthesized speech by the mean opinion score (MOS) test method. The subjects were 10 Japanese students belonging to our research group. Twenty sentences were chosen at random from the evaluation sentences. Samples were presented in random order for each test sentence. In the MOS test, after listening to each test sample, the subjects were asked to assign the sample a five-point naturalness score (5: natural – 1: poor).

4.3.2 Experimental results

Comparison of HSMM with HMM

This experiment compared the following four models.

- **ML-HMM** : The HMMs were trained by the ML criterion. Model structures were selected by the MDL criterion.
- **ML-HSMM** : The HSMMs were trained by the ML criterion. Model structures were selected by the MDL criterion.
- **Bayes-HMM** : The HMMs were trained by the Bayesian method. Model structures were selected by the Bayesian criterion with cross validation.
- **Bayes-HSMM** : The HSMMs were trained by the Bayesian method. Model structures were selected by the Bayesian criterion with cross validation.

Table 4.1: Number of states of selected model structure by the conventional and proposed methods.

	mel-cepstram	F_0	duration
ML-HMM	1,115	2,267	275
ML-HSMM	1,128	2,272	283
Bayes-HMM	9,532	16,044	3,005
Bayes-HSMM	9,485	16,130	3,490

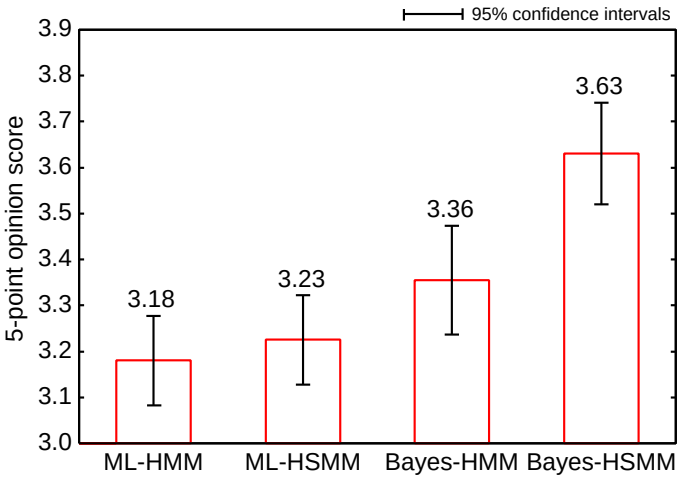

Figure 4.1: Mean opinion scores of speech synthesized by the conventional and proposed methods. Error bars show 95% confidence intervals.

Table 4.1 represents the details of the number of states.

Figure 4.1 plots the experimental results. It can be seen from the figure that the proposed model **Bayes-HSMM** achieved a better subjective score than the conventional model **Bayes-HMM**, and the subjective score of **ML-HSMM** was better than **ML-HMM**. Consequently, the speech quality is improved by using HSMMs as the acoustic models. Moreover, the proposed model **Bayes-HSMM** outperformed the model **ML-HSMM**. These results clearly show the effectiveness of the proposed model. The number of states of **Bayes-HMM** and **Bayes-HSMM** was considerable larger than **ML-HMM** and **ML-HSMM**. Although the large model structure alleviated the over-smoothing problem, the ML training leads to the over-fitting problem. However, the Bayesian approach avoided the over-fitting problem because the posterior distributions of the model parameters were used. Therefore, the Bayesian approach overcame the over-fitting and over-smoothing problems simultaneously. Consequently, most of the subjects observed that the proposed model improved the naturalness in spectrum and excitation.

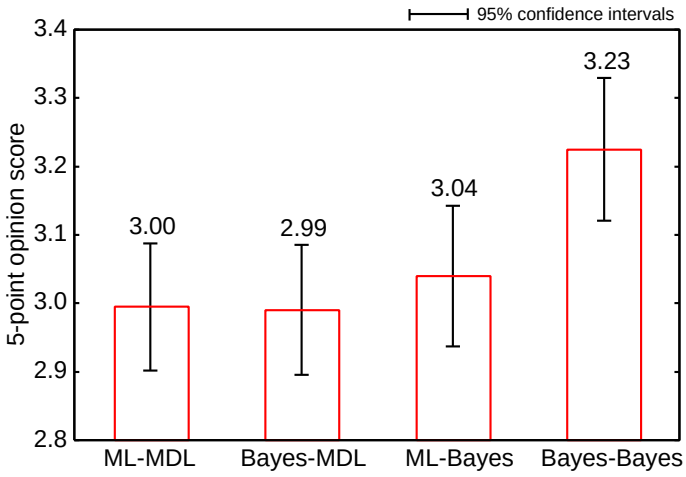

Figure 4.2: Mean opinion scores of speech synthesized by the conventional, proposed and swapped models. Error bars show 95% confidence intervals.

Comparison of Model Structures

In this experiment, the model structures of **ML-HSMM** and **Bayes-HSMM** were swapped in order to compare the effect of model structures. Therefore, the following four models were compared.

- **ML-MDL** : The HSMMs were trained by the ML criterion and model structures were selected by the MDL criterion. This is the same model as **ML-HSMM**.
- **Bayes-Bayes** : The HSMMs were trained by the Bayesian criterion and model structures were selected by the Bayesian criterion with cross validation. This is the same model as **Bayes-HSMM**.
- **Bayes-MDL** : The HSMMs were trained by the Bayesian criterion and the model structures of **ML-HSMM** were used.
- **ML-Bayes** : The HSMMs were trained by the ML criterion and the model structures of **Bayes-HSMM** were used.

Figure 4.2 shows the results of the subjective listening test. It can be seen from the figure that the proposed method **Bayes-Bayes** achieved a better subjective score than the conventional method **ML-MDL**. Moreover, although **Bayes-MDL** is trained by the Bayesian criterion, the subjective score of **Bayes-MDL** was worse than **Bayes-Bayes**, and although **ML-Bayes** has the similar number of states as **Bayes-Bayes**, the subjective score of **ML-Bayes** was worse than **Bayes-Bayes**. Because the model structure of **ML-Bayes** is too

big for the ML training, **ML-Bayes** leads to the over-fitting problem. Thus, the error bar of **ML-Bayes** in Figure 4.2 are larger than others. These results clearly show the effectiveness of the proposed method in both the model training and model structure selection. Most of the subjects observed that the proposed method improved the naturalness in spectrum and excitation.

4.4 Bayesian speech synthesis integrating training and synthesis processes

In Bayesian speech synthesis, the estimation of the posterior distributions, model selection, and speech parameter generation are consistently performed by maximizing the log marginal likelihood. The posterior distributions of all variables are obtained by using the VB method. Then, the obtained posterior distribution of the model parameters depends on not only the training data, but also the synthesis data. In a basic speech synthesis situation, the observed data for the synthesis sentences is not given beforehand. Therefore, the posterior distributions cannot be obtained. To overcome this problem, it typically assumes that the posterior distribution of the model parameters is independent of the synthesis data [18, 19]. As a result of this approximation, the Bayesian speech synthesis system is separated into training and synthesis parts, as the conventional ML-based system, and the posterior distribution of the model parameters and decision trees can be obtained from only the training data. However, although the posterior distributions can be estimated, they don't consider synthesis data, and the system doesn't represent the Bayesian speech synthesis exactly. This section proposes a speech synthesis technique integrating training and synthesis processes based on the Bayesian framework. This method removes the approximation and leads to an algorithm that the posterior distributions, decision trees and synthesis data are iteratively updated.

4.4.1 Speech parameter generation

In the synthesis part of HMM-based speech synthesis, first, an arbitrarily given text to be synthesized is converted into a context-dependent label sequence and a sentence HMM is constructed by concatenating context-dependent HMMs according to the label sequence. Second, the optimal state sequence of the sentence HMM is determined. Third, a speech parameter sequence is generated for a given state sequence. From Eq. (4.3), the optimal speech parameter sequence for Bayesian speech synthesis can be generated by maximizing the marginal likelihood. Thus, the optimal speech parameter sequence $\hat{\boldsymbol{o}}$ can be generated by maximizing the lower bound $\mathcal{F}$ in Eq. (4.5) because the VB method guarantees that the log marginal likelihood is approximately the lower bound $\mathcal{F}$.

$$\begin{aligned} \hat{\boldsymbol{o}}_{Bayes} &= \arg\max_{\boldsymbol{o}} \log P(\boldsymbol{o}, \boldsymbol{O} \mid s, S) \\ &\approx \arg\max_{\boldsymbol{o}} \mathcal{F} \end{aligned} \tag{4.38}$$

We assume that a speech parameter vector $\boldsymbol{o}_t$ consists of a static feature vector $\boldsymbol{c}_t$ and its first and second order dynamic feature vectors.

$$\begin{aligned} \boldsymbol{o} &= \boldsymbol{W}\boldsymbol{c} \\ &= \left[(\boldsymbol{W}\boldsymbol{c})_1^\top, (\boldsymbol{W}\boldsymbol{c})_2^\top, \ldots, (\boldsymbol{W}\boldsymbol{c})_T^\top\right]^\top \qquad (4.39) \\ (\boldsymbol{W}\boldsymbol{c})_t &= \left[\boldsymbol{c}_t^\top, \Delta\boldsymbol{c}_t^\top, \Delta^2\boldsymbol{c}_t^\top\right]^\top \qquad (4.40) \end{aligned}$$

where $\boldsymbol{W}$ is a window matrix to calculate dynamic features from static features [51]. The dynamic feature vectors are automatically determined from the window matrix $\boldsymbol{W}$ and the static feature sequence. Consequently, only a static feature vector sequence $\boldsymbol{c}$ is estimated in the synthesis part. From Eq. (4.38), the optimal static feature sequence $\hat{\boldsymbol{c}}$ is generated by maximizing the lower bound $\mathcal{F}$. Moreover, under the condition of Eq. (4.39), the optimal static feature sequence $\hat{\boldsymbol{c}}$ can be determined by solving the following equation:

$$\begin{aligned} \frac{\partial \mathcal{F}}{\partial \boldsymbol{c}} &= \frac{\partial}{\partial \boldsymbol{c}} \left\langle \log \frac{P(\boldsymbol{W}\boldsymbol{c}, \boldsymbol{z}, \boldsymbol{O}, \boldsymbol{Z}, \boldsymbol{\Lambda} \mid s, S)}{Q(\boldsymbol{z})Q(\boldsymbol{Z})Q(\boldsymbol{\Lambda})} \right\rangle_{Q(\boldsymbol{z})Q(\boldsymbol{Z})Q(\boldsymbol{\Lambda})} \\ &= \boldsymbol{0} \qquad (4.41) \end{aligned}$$

In the Bayesian speech synthesis framework, the estimation of the posterior distributions, model selection, and speech parameter generation consistently maximize the lower bound $\mathcal{F}$.

4.4.2 Approximation for estimating posterior distributions

The obtained posterior distribution of model parameters $Q(\boldsymbol{\Lambda})$ in Eq. (4.9) depend on not only the training data $\boldsymbol{O}$, but also the synthesis data $\boldsymbol{o}$. However, in a basic speech synthesis situation, the observed data of synthesis sentences is not given previously. Therefore, the posterior distributions represented in Eq. (4.9) cannot be estimated. To overcome this problem, one typically assumes that the posterior distribution of the model parameters is independent of the synthesis data [18,19]. The lower bound of the log marginal likelihood with respect to only the training data $\boldsymbol{O}$ can be represented as follows.

$$\begin{aligned} \log P(\boldsymbol{O} \mid S) &= \log \sum_{\boldsymbol{Z}} \int P(\boldsymbol{O}, \boldsymbol{Z}, \boldsymbol{\Lambda} \mid S) d\boldsymbol{\Lambda} \\ &\geq \left\langle \log \frac{P(\boldsymbol{O}, \boldsymbol{Z}, \boldsymbol{\Lambda} \mid S)}{\bar{Q}(\boldsymbol{Z})\bar{Q}(\boldsymbol{\Lambda})} \right\rangle_{\bar{Q}(\boldsymbol{Z})\bar{Q}(\boldsymbol{\Lambda})} \\ &= \bar{\mathcal{F}} \qquad (4.42) \end{aligned}$$

The posterior distributions $\bar{Q}(\boldsymbol{Z})$ and $\bar{Q}(\boldsymbol{\Lambda})$ can be estimated by maximizing the lower bound $\bar{\mathcal{F}}$. The posterior distribution of the model parameters $\bar{Q}(\boldsymbol{\Lambda})$ is represented as

follows.

$$\bar{Q}(\boldsymbol{\Lambda}) = \bar{C}_{\boldsymbol{\Lambda}} P(\boldsymbol{\Lambda}) \exp \langle \log P(\boldsymbol{O}, \boldsymbol{Z} \mid S, \boldsymbol{\Lambda}) \rangle_{\bar{Q}(\boldsymbol{Z})} \tag{4.43}$$

Equation (4.43) indicates that the posterior distribution $\bar{Q}(\boldsymbol{\Lambda})$ is independent of the synthesis data and that it can be estimated by using only the training data. Since the same approximation is used in the Bayesian model selection, the optimal decision trees are selected by maximizing the lower bound $\bar{\mathcal{F}}$ instead of $\mathcal{F}$.

$$\hat{m} = \arg\max_{m} \bar{\mathcal{F}} \tag{4.44}$$

Consequently, the decision trees are selected independently of the synthesis data. Additionally, Eq. (4.41) can be represented by the estimated posterior distribution $\bar{Q}(\boldsymbol{\Lambda})$ and the determined state sequence as follows.

$$\left\langle \frac{\partial}{\partial \boldsymbol{c}} \log P(\boldsymbol{W}\boldsymbol{c} \mid \boldsymbol{z}, \boldsymbol{\Lambda}) \right\rangle_{\bar{Q}(\boldsymbol{\Lambda})} = \boldsymbol{0} \tag{4.45}$$

Equation (4.45) can be solved efficiently by using the Cholesky or QR decomposition [51]. Therefore, the computational cost is almost the same as the ML criterion.

The approximation that the posterior distribution of the model parameters is independent of the synthesis data $\boldsymbol{o}$ enables the Bayesian speech synthesis system to be separated into training and synthesis parts as the conventional ML-based system and to obtain the posterior distribution of model parameters and decision trees from only the training data. However, although the posterior distributions can be estimated, they don't take into account synthesis data, and the system doesn't represent the Bayesian speech synthesis exactly. To overcome this problem, this chapter proposes a speech synthesis technique integrating training and synthesis processes based on the Bayesian framework.

4.4.3 Integration of training and synthesis processes

The proposed method removes the approximation and derives an algorithm that the posterior distributions, decision trees, and synthesis data are iteratively updated. In the proposed framework, the generated speech parameters of the synthesis sentences are used instead of the observed data. That is, the posterior distributions and decision trees are estimated from the training data and the generated speech parameters, and the speech parameters are generated from the estimated posterior distributions. Since the posterior distributions, decision trees, and generated speech parameters depend on each other, they are iteratively updated as the EM algorithm. Initial synthesis data are generated by using

the framework described in the preceding section 4.4.2. Once the generated speech parameters are obtained, they can be used for estimating the posterior distribution. The new lower bound with the generated speech parameters is defined as follows.

$$\begin{aligned}\log P(\tilde{\boldsymbol{o}}, \boldsymbol{O} \mid s, S) &= \log \sum_{\tilde{\boldsymbol{z}}} \sum_{\boldsymbol{Z}} \int P(\tilde{\boldsymbol{o}}, \tilde{\boldsymbol{z}}, \boldsymbol{O}, \boldsymbol{Z}, \boldsymbol{\Lambda} \mid s, S) d\boldsymbol{\Lambda} \\ &\geq \left\langle \log \frac{P(\tilde{\boldsymbol{o}}, \tilde{\boldsymbol{z}}, \boldsymbol{O}, \boldsymbol{Z}, \boldsymbol{\Lambda} \mid s, S)}{\tilde{Q}(\tilde{\boldsymbol{z}}) \tilde{Q}(\boldsymbol{Z}) \tilde{Q}(\boldsymbol{\Lambda})} \right\rangle_{\tilde{Q}(\tilde{\boldsymbol{z}}) \tilde{Q}(\boldsymbol{Z}) \tilde{Q}(\boldsymbol{\Lambda})} \\ &= \tilde{\mathcal{F}} \end{aligned} \tag{4.46}$$

where $\tilde{\boldsymbol{o}}$ is the generated speech parameter sequence. By maximizing the lower bound $\tilde{\mathcal{F}}$, the posterior distribution can be estimated in the same fashion as Eq. (4.9).

$$\begin{aligned}\tilde{Q}(\boldsymbol{\Lambda}) &= \tilde{C}_{\boldsymbol{\Lambda}} P(\boldsymbol{\Lambda}) \exp \left\langle \log P(\tilde{\boldsymbol{o}}, \tilde{\boldsymbol{z}} \mid s, \boldsymbol{\Lambda}) \right\rangle_{\tilde{Q}(\tilde{\boldsymbol{z}})} \\ &\quad \times \exp \left\langle \log P(\boldsymbol{O}, \boldsymbol{Z} \mid S, \boldsymbol{\Lambda}) \right\rangle_{\tilde{Q}(\boldsymbol{Z})} \end{aligned} \tag{4.47}$$

The posterior distributions are estimated from the training data and the generated speech parameters instead of the observed speech parameters. Additionally, the decision trees are selected by maximizing the lower bound $\tilde{\mathcal{F}}$.

$$\hat{m} = \arg\max_{m} \tilde{\mathcal{F}} \tag{4.48}$$

Equation (4.41) can be represented by the estimated posterior distribution $\tilde{Q}(\boldsymbol{\Lambda})$ and the determined state sequence.

$$\left\langle \frac{\partial}{\partial \boldsymbol{c}} \log P(\boldsymbol{W}\boldsymbol{c} \mid \boldsymbol{z}, \boldsymbol{\Lambda}) \right\rangle_{\tilde{Q}(\boldsymbol{\Lambda})} = \boldsymbol{0} \tag{4.49}$$

In the proposed framework, the estimation of posterior distributions, model selection and speech parameter generation consistently maximize the lower bound $\tilde{\mathcal{F}}$. The posterior distributions, decision trees, and synthesis data are iteratively updated. The iterative process is as follows.

1. Initial speech parameters of synthesis sentences are generated with in the represented framework (Eq. (4.45)).
2. The posterior distributions $\tilde{Q}(\tilde{\boldsymbol{z}}) \tilde{Q}(\boldsymbol{Z}) \tilde{Q}(\boldsymbol{\Lambda})$ and decision trees are re-estimated by maximizing the lower bound $\tilde{\mathcal{F}}$ (Eqs. (4.47) and (4.48)).
3. Speech parameters of synthesis sentences are re-generated by using the estimated posterior distribution (Eq. (4.49)).

4. Steps 2 and 3 are iterated until the value of $\tilde{\mathcal{F}}$ converge.

Although the iterative process increase the computational cost, the final posterior distributions is more appropriate than one used in the previous method for synthesis sentences.

The key question is *how many synthesis sentences should be used for estimating the posterior distributions?* Here, we discuss two approaches about the number of synthesis sentences.

- **Sentence:** The generated speech parameters of one synthesis sentence are used as $\tilde{\boldsymbol{o}}$.
- **Batch:** The generated speech parameters of all synthesis sentences are used as $\tilde{\boldsymbol{o}}$.

Sentence estimates different posterior distributions and model structures for each synthesis sentence. On the other hand, **Batch** estimates the same posterior distributions and model structures for all synthesis sentences. Therefore, **Sentence** needs the larger computational cost than **Batch**.

4.5 Experiments

4.5.1 Experimental conditions

The experiments used the ATR Japanese speech database [53] B-set, which consists of 503 phonetically balanced sentences. The first 450 of the 503 sentences, uttered by one male speaker (MHT), were used for training. The remaining 53 sentences were used for the evaluations. Speech signals were sampled at a rate of 16 kHz and windowed at a 5-ms frame rate using a 25-ms Blackman window. Feature vectors consisted of spectrum and F_0 parameter vectors. The spectrum parameter vectors consisted of 24 mel-cepstral coefficients, and their delta and delta-delta coefficients. The F_0 parameter vectors consisted of log F_0 and its delta and delta-delta. A five-state, left-to-right MSD-HSMM [52, 54] without skip transitions was used. Each state output PDF was composed of spectrum and F_0 streams. The spectrum stream was modeled by single multi-variate Gaussian distributions with diagonal covariance matrices. The F_0 stream was modeled by a multi-space probability distribution consisting of a Gaussian distribution for voiced frames and a discrete distribution for unvoiced frames. Each state duration PDF was modeled by a one-dimensional Gaussian distribution. The decision tree-based context clustering technique was separately applied to distributions of spectrum, F_0, and state duration.

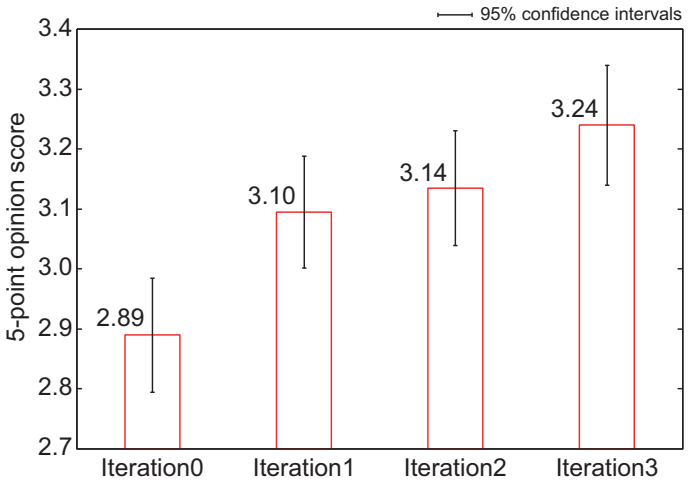

Figure 4.3: Mean opinion scores of speech synthesized by the baseline and proposed methods. Error bars show 95% confidence intervals.

A subjective listening test was conducted to evaluate the quality of the synthesized speech. The test assessed the naturalness of the converted speech by the mean opinion score (MOS) test method. The subjects were 10 Japanese students belonging to our research group. Twenty sentences were chosen at random from the evaluation sentences. Samples were presented in random order for each synthesis sentence. In the MOS test, after listening to each test sample, the subjects were asked to assign the sample a five-point naturalness score (5: natural – 1: poor).

4.5.2 Comparing the number of updates

This experiment evaluated the effectiveness of the proposed iterative updates by comparing the following four systems.

- **Iteration0** : The posterior distributions were trained from only the training data.
- **Iteration1** : The posterior distributions were trained from the training data and the speech parameters generated by **Iteration0**.
- **Iteration2** : The posterior distributions were trained from the training data and the speech parameters generated by **Iteration1**.
- **Iteration3** : The posterior distributions were trained from the training data and the speech parameters generated by **Iteration2**.

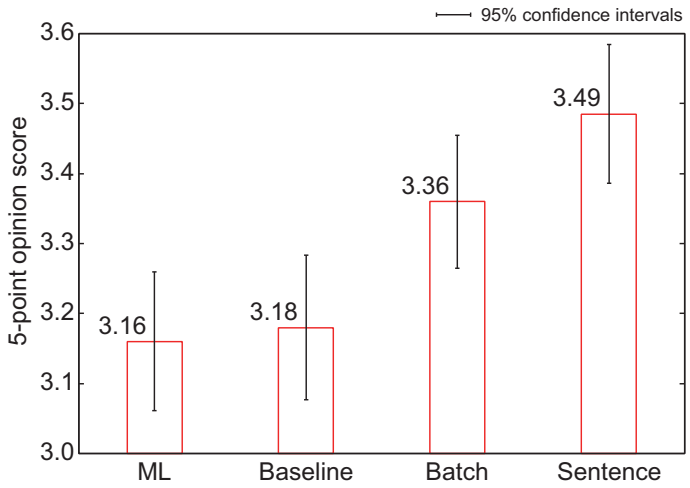

Figure 4.4: Mean opinion scores of speech synthesized by the baseline and proposed methods. Error bars show 95% confidence intervals.

Iteration0 was the baseline Bayesian speech synthesis system described in Section 4.4.2. **Iteration1**, **Iteration2**, and **Iteration3** were the proposed system integrating training and synthesis processes described in Section 4.4.3, and they were based on sentence-form integration. In each iteration, the posterior distributions were updated five times. Therefore, in this experiment, the number of updates was different for each system.

Figure 4.3 plots the experimental results. Although there were not confidence intervals, it is clear that the subjective score increased as the number of training iterations increased. These results clearly show the effectiveness of the training and synthesis iterations. The decision trees constructed in the context clustering varied between the four systems. This shows that the posterior distributions were optimized as a result of integrating the training and synthesis processes.

4.5.3 Comparing systems

This experiment compared the following four systems.

- **ML** : The conventional ML-based speech synthesis system. The HMMs were trained by using the ML criterion. The decision trees were selected by the MDL criterion [44].
- **Baseline** : The baseline Bayesian speech synthesis system described in Section 4.4.2.
- **Batch** : The proposed Bayesian speech synthesis system based on the batch-form integration described in Section 4.4.3.

- **Sentence** : The proposed Bayesian speech synthesis system based on the sentence-form integration described in Section 4.4.3. This system was the same as **Iteration3** of the previous experiment.

The computational costs of **ML**, **Baseline**, and **Batch** were almost same because the number of updates was same in this experiment. However, since **Sentence** estimated different posterior distributions and model structures for each synthesis sentence, the computational cost was 53 times as large as **Batch**.

Figure 4.4 shows the results of the subjective listening test. **Baseline** was better than **ML**, although the gain was not significant. In addition, **Batch** and **Sentence** outperformed **Baseline**. These performance gains illustrate the effectiveness of the proposed Bayesian speech synthesis framework integrating training and synthesis processes. The figure also shows that **Sentence** performed better than **Batch**. Although **Batch** used all generated synthesis data to estimate the posterior distributions, the posterior distributions and model structures of **Batch** were common for all synthesis sentences. In contrast, **Sentence** estimated different posterior distributions and model structures for each synthesis sentence. The experimental results illustrate that the quality of the synthesized speech improved when the posterior distributions were optimized for each synthesis sentence.

4.6 Summary

This chapter proposed the new framework of speech synthesis based on the Bayesian approach. In the proposed framework, all processes for constructing the system could be derived from one single predictive distribution which represents the problem of speech synthesis directly. The results on the MOS test demonstrated that the proposed method outperform the conventional one.

Furthermore, this chapter also proposed a speech synthesis technique integrating training and synthesis processes based on the Bayesian framework. The proposed method removes the approximation that the posterior distribution of the model parameters is independent of the synthesis data and derives an algorithm that the posterior distributions, decision trees and synthesis data are iteratively updated. Both sentence-form and batch-form integrations were tested. The sentence-form integration estimates different posterior distributions and decision trees for each synthesis sentence, whereas the batch-form integration estimates the same ones for all synthesis sentences. The results of MOS synthesis demonstrated that the proposed method outperforms the baseline method and the sentence-form integration performed better than the batch-form integration.

Our future work will include investigation of the relation between the amount of training data and the quality of speech synthesized by the proposed method.

Chapter 5

An analysis of machine translation and speech synthesis in speech-to-speech translation system

In speech-to-speech translation (S2ST), the source language speech is translated into target language speech. A S2ST system can help to overcome the language barrier, and is essential for providing more natural interaction. A S2ST system consists of three components: speech recognition, machine translation and speech synthesis. In the simplest S2ST system, only the single-best output of one component is used as input to the next component. Therefore, errors of the previous component strongly affect the performance of the next component. Due to errors in speech recognition, the machine translation component cannot achieve the same level of translation performance as achieved for correct text input. To overcome this problem, many techniques for integration of speech recognition and machine translation have been proposed, such as [20, 21]. In these, the impact of speech recognition errors on machine translation is alleviated by using N-best list or word lattice output from the speech recognition component as input to the machine translation component. Consequently, these approaches can improve the performance of S2ST significantly. However, the speech synthesis component is not usually considered. The output speech for translated sentences is generated by the speech synthesis component. If the quality of synthesized speech is bad, users will not understand what the system said: the quality of synthesized speech is obviously important for S2ST and any integration method intended to improve the end-to-end performance of the system should take account of the speech synthesis component.

The EMIME project [55] is developing personalized S2ST, such that the a user's speech input in one language is used to produce speech output in another language. Speech char-

acteristics of the output speech are adapted to the input speech characteristics using cross-lingual speaker adaptation techniques [56]. While personalization is an important area of research, this chapter focuses on the impact of the machine translation and speech synthesis components on end-to-end performance of an S2ST system. In order to understand the degree to which each component affects performance, we investigate integration methods. We first conducted a subjective evaluation divided into three sections: speech synthesis, machine translation, and speech-to-speech translation. Various translated sentences were evaluated by using N-best translated sentences output from the machine translation component. The individual impacts of the machine translation and the speech synthesis components are analyzed from the results of this subjective evaluation.

5.1 Related work

In the field of spoken dialog systems, the quality of synthesized speech is one of the most important features because users cannot understand what the system said if the quality of synthesized speech is low. Therefore, integration of natural language generation and speech synthesis has been proposed [57–59].

In [57], a method was proposed for integration of natural language generation and unit selection based speech synthesis which allows the choice of wording and prosody to be jointly determined by the language generation and speech synthesis components. A template-based language generation component passes a word network expressing the same content to the speech synthesis component, rather than a single word string. To perform the unit selection search on this word network input efficiently, weighted finite-state transducers (WFSTs) are employed. The weights of the WFST are determined by join costs, prosodic prediction costs, and so on. In an experiment, this system achieved higher quality speech output. However, this method cannot be used with most existing speech synthesis systems, because they do not accept word networks as input.

An alternative to the word network approach is to re-rank sentences from the N-best output of the natural language generation component [58]. N-best output can be used in conjunction with any speech synthesis system although the natural language generation component must be able to construct N-best sentences. In this method, a re-ranking model selects the sentences that are predicted to sound most natural when synthesized with the unit selection based speech synthesis component. The re-ranking model is trained from the subjective scores of the synthesized speech quality assigned in a preliminary evaluation and features from the natural language generation and speech synthesis components such as word N-gram model scores, join cost, and prosodic prediction costs. Experimen-

tal results demonstrated higher quality speech output. Similarly, a re-ranking model for N-best output was also been proposed in [59]. In contrast to [58], this model used a much smaller data set for training and a larger set of features, but reached the same performance as reported in [58].

These are integration methods for natural language generation and speech synthesis for spoken dialog systems. In contrast to these methods, our focus is on the integration of machine translation and speech synthesis for S2ST. To this end, we first conducted a subjective evaluation – using Amazon Mechanical Turk [60] – then analyzed the impact of machine translation and speech synthesis on S2ST.

5.2 Subjective evaluation

5.2.1 Systems

In the subjective evaluation, a Finnish-to-English S2ST system was used. To focus on the impacts of machine translation and speech synthesis, the correct sentences were used as the input of the machine translation component instead of the speech recognition results.

The system developed in [61] was used as the machine translation component of our S2ST system. This system is *HiFST*: a hierarchical phrase-based system implemented with weighted finite-state transducers [62]. 865,732 parallel sentences from the EuroParl corpus [63] were used as training data, and 3,000 parallel sentences from the same corpus was used as development data. When the system was evaluated on 3,000 sentences in [61], it obtained 28.9 on the BLEU-4 measure.

As the speech synthesis component, an HMM-based speech synthesis system (HTS) [64] was used. 8,129 sentences uttered by one male speaker were used for training acoustic models. Speech signals were sampled at a rate of 16 kHz and windowed by an F_0-adaptive Gaussian window with a 5 ms shift. Feature vectors comprised 138-dimensions: 39-dimension STRAIGHT [65] mel-cepstral coefficients (plus the zero-th coefficient), log F_0, 5 band-filtered aperiodicity measures, and their dynamic and acceleration coefficients. We used 5-state left-to-right context-dependent multi-stream MSD-HSMMs [52,54]. Each state had a single Gaussian. Festival [66] was used for deriving full-context labels from the text; the labels include phoneme, part of speech (POS), intonational phrase boundaries, pitch accent, and boundary tones.

The test data comprised 100 sentences from EuroParl corpus not included in the machine translation training data. The machine translation component output the 20-best transla-

Table 5.1: Example of N-best MT output texts

N	Output text
Reference	We can support what you said.
1	We support what you have said.
2	We support what you said.
3	We are in favour of what you have said.
4	We support what you said about.
5	We are in favour of what you said.
6	We support what you have said about.
7	We will support what you have said.
8	We support what you have just said.
9	We support what you say.
10	We support that what you have said.
11	We will support what you said.
12	Support what you have said.
13	We support it, what you have said.
14	We are in favour of what you have just said.
15	We are in favour of what you said about.
16	We will support what you said about.
17	We will support what you have said about.
18	Support what you said about.
19	We will support what you have just said.
20	We will support what you say.

tions for each input sentence, resulting in 2,000 translated sentences. To these, we added reference translations to give a total of 2,100 sentences to use in the evaluation. Table 5.1 shows an example of top 20-best translated sentences.

5.2.2 Evaluation procedure

The evaluation comprised 3 sections: In section 1, speech synthesis was evaluated. Evaluators listened to synthesized speech and assigned scores for naturalness (**TTS**). We asked evaluators to assign a score without considering the correctness of grammar or content. In section 2, speech-to-speech translation was evaluated. Evaluators listened to synthesized speech, then typed in the sentence; we measured their word error rate (**WER**). After this, evaluators assigned scores for "Adequacy" and "Fluency" of the typed-in sentence (**S2ST-Adequacy** and **S2ST-Fluency**). Here, "Adequacy" indicates how much of the information from the reference translation sentence was expressed in the sentence and "Fluency"

Table 5.2: Correlation coefficients between **TTS** or **WER** and **MT** scores

	MT-Adequacy	**MT-Fluency**
TTS	0.12	0.24
WER	-0.17	-0.25

indicates that how fluent the sentence was [67]. These definitions were provided to the evaluators. "Adequacy" and "Fluency" measures do not need bilingual evaluators; they can be evaluated by monolingual target language listeners. These measures are widely used in machine translation evaluations, e.g., conducted by NIST and IWSLT. In section 3, machine translation was evaluated. Evaluators didn't listen to synthesized speech. They read translated sentences and assigned scores of "Adequacy" and "Fluency" for each sentence (**MT-Adequacy** and **MT-Fluency**).

TTS, **S2ST-Adequacy**, **S2ST-Fluency**, **MT-Adequacy**, and **MT-Adequacy** were evaluated on five-point mean opinion score (MOS) scales. Evaluators assigned scores to 42 test sentences in each section. 150 people participated in the evaluation.

5.2.3 Impact of MT and WER on S2ST

First, we analyzed the impact of the translated sentences and the intelligibility of synthesized speech on S2ST. **WER** averaged across all test samples was 6.49%. The correlation coefficients between **MT-Adequacy** and **S2ST-Adequacy** and between **MT-Fluency** and **S2ST-Fluency** were strong (0.61 and 0.68, respectively).

The correlation coefficient between **WER** and **S2ST-Adequacy** was -0.21, and the correlation coefficient between **WER** and **S2ST-Fluency** was -0.20. These are only weak correlations. The impact of the translated sentences on S2ST is larger than the impact of the intelligibility of the synthesized speech, although this does affect the performance of S2ST.

5.2.4 Impact of MT on TTS and WER

Next, we analyzed the impact of the translated sentences on the naturalness and intelligibility of synthesized speech. Table 5.2 shows the correlation coefficients between **TTS** and **MT** scores, and the correlation coefficients between **WER** and **MT** scores. **MT-Fluency** has a stronger correlation with both **TTS** and **WER** than **MT-Adequacy**. That is, the naturalness and intelligibility of synthesized speech were more affected by the

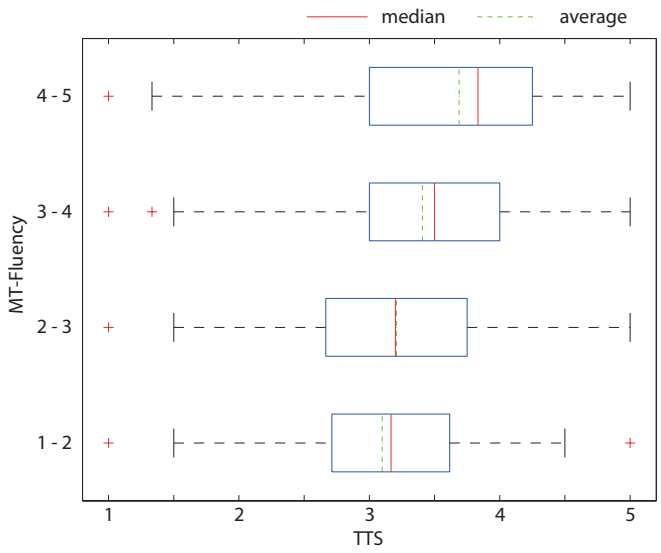

Figure 5.1: Boxplots of **TTS** divided into four groups by **MT-Fluency**

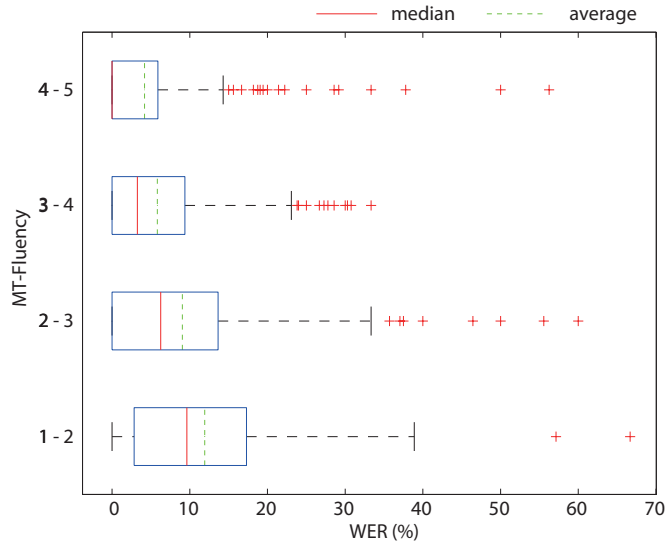

Figure 5.2: Boxplots of **WER** divided into four groups by **MT-Fluency**

fluency of the translated sentences than by the content of them. Therefore, next we focused on the relationship between the fluency of the translation output and the synthesized speech.

Figure 5.1 shows boxplots of **TTS** divided into four groups by **MT-Fluency**. In this figure, the median and average scores are also shown. This figure illustrates that the median and average scores of **TTS** are slightly improved by increasing **MT-Fluency**. This is

Table 5.3: Table of correlation coefficients between **MT-Fluency** and word N-gram score

1-gram	2-gram	3-gram	4-gram	5-gram
0.28	0.39	0.42	0.43	0.44

presumed to be because the speech synthesis text processor (Festival, in our case) often produced incorrect full-context labels due to the errors in syntactic analysis of disfluent and ungrammatical translated sentences. In addition, the psychological effect called "Llewelyn reaction" appears to affect the results. The "Llewelyn reaction" is that evaluators perceive lower speech quality when the sentences are less fluent or the content of the sentences is less natural, even if the actual quality of synthesized speech is same. Therefore, we conclude that the speech synthesis component will tend to generate more natural speech as the translated sentences become more fluent. Figure 5.2 shows the boxplots of **WER** divided into four groups by **MT-Fluency**. From this figure, it can be seen that the median and average scores of **WER** improve and the variance of boxplots shrinks, with increasing **MT-Fluency**. This is presumed to be because evaluators can predict the next word when the translated sentence does not include unusual words or phrases, in addition to the naturalness of synthesized speech being better when the sentences were more fluent, as previously described. Therefore, the intelligibility of synthesized speech is improved as the translated sentences become more fluent, even though all sentences are synthesized by the same system.

5.2.5 Correlation between MT Fluency and N-gram scores

We have shown that the naturalness and intelligibility of the synthesized speech are strongly affected by the fluency of sentences. It is well known in the field of machine translation that the fluency of translated sentences can be improved by using long-span word-level N-grams. Therefore, we computed the correlation coefficient between **MT-Fluency** and word N-gram score. The word N-gram models we used were created using the SRILM toolkit [33], from the same English sentences used for training the machine translation component. Kneser-Ney smoothing was employed.

Table 5.3 shows the correlation coefficient between **MT-Fluency** and word N-gram score. The word 5-gram gave the strongest correlation coefficient of 0.44. Although there were weak correlations between **MT-Fluency** and word N-gram score on raw data, it was difficult to find strong correlation coefficients. Therefore, **MT-Fluency** scores were divided into 200 bins according to the word 5-gram score and subsequently average **MT-Fluency** scores for each bin were computed. In Figure 5.3, the averaged **MT-Fluency** scores and

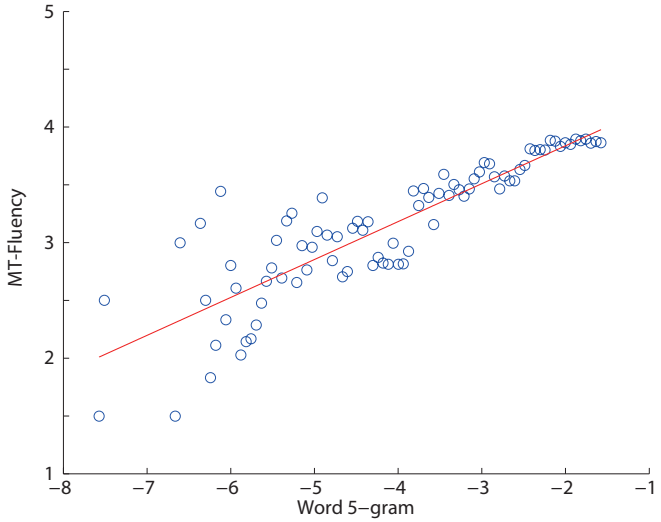

Figure 5.3: Correlation between **MT-Fluency** and word 5-gram score

word 5-gram scores are shown, and the regression line is illustrated by the line. Now, the correlation coefficient is 0.87. This result indicates that the word 5-gram score is an appropriate feature for measuring the average perceived fluency of translated sentences.

5.2.6 Correlation between TTS and N-gram scores

P.563 is an objective measure for predicting the quality of natural speech in telecommunication applications [68]. However, we found no correlation between **TTS** and P.563. So, we looked for correlations with other objective measures. It is well known that speech synthesis systems generally produce better quality speech when the input sentence is in-domain (i.e., similar to sentences found in the training data). Therefore, we computed the correlation coefficient between **TTS** and phoneme N-gram score of the sentence being synthesized; the N-gram score is a measure of the coverage provided by the training data for that particular sentence. The phoneme N-gram model was estimated from the English sentences used for training the speech synthesizer. Table 5.4 shows the correlation coefficients of **TTS** and phoneme N-gram scores; the 4-gram model gave the strongest correlation coefficient of 0.20. Figure 5.4 shows the bin-averaged **TTS** scores and phoneme 4-gram scores. Now, the correlation coefficient is 0.81. Although the correlation between **TTS** and phoneme N-gram score was weak on the raw data, there is a strong correlation between bin-averaged **TTS** and phoneme N-gram score. This result suggests that the phoneme 4-gram score is a good predictor of the expected naturalness of synthesized speech.

Table 5.4: Table of correlation coefficients between **TTS** and phoneme N-gram score

1-gram	2-gram	3-gram	4-gram	5-gram
0.05	0.15	0.19	0.20	0.18

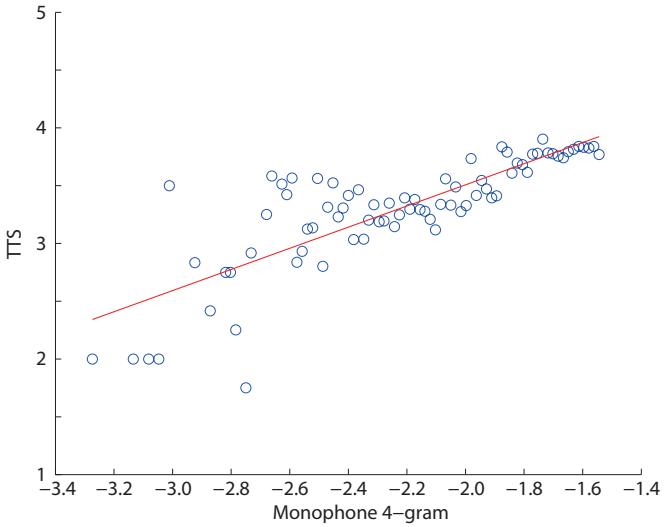

Figure 5.4: Correlation between **TTS** and phoneme 4-gram score

The ability to predict synthetic speech naturalness before generating the speech could be used in other applications, such as sentence selection (as in this work, or in natural language generation with speech output), voice selection before generating speech. We hope to investigate this further in the future.

5.3 Summary

This chapter has provided an analysis of the impacts of machine translation and speech synthesis on speech-to-speech translation. It has been shown that the naturalness and intelligibility of the synthesized speech are strongly affected by the fluency of the translated sentences. The intelligibility of synthesized speech is improved as the translated sentence become more fluent. In addition, it was found that long-span word N-gram scores correlate well with the perceived fluency of sentences and that phoneme N-gram scores correlate well with the perceived naturalness of synthesized speech. Our future work will include investigations into the integration of machine translation and speech synthesis using word N-gram and phoneme N-gram scores.

Chapter 6

Conclusions

The present paper described improved statistical models for speech-to-speech translation. In Chapter 2, a reordering model using a source-side parse-tree for phrase-based statistical machine translation was proposed. The proposed model is an extension of the IST-ITG constraints. Both the IST-ITG constraints and the proposed reordering model fix the phrase position for the global reorderings. However, the proposed method can conduct a probabilistic evaluation of target word reorderings which the IST-ITG constraints cannot. In E-J and E-C translation experiments, the proposed method resulted in a 0.49-point improvement (29.31 to 29.80) and a 0.33-point improvement (18.60 to 18.93) in word BLEU-4 compared with IST-ITG constraints, respectively. This indicates the validity of the proposed reordering model. Future work will focus on a simultaneous training of translation and reordering models. Moreover, we will deal with difference between source and target tree structures in multi level like in [40]. In Chapter 3, the Bayesian context clustering using cross validation for speech recognition was proposed. In the proposed method, the prior distributions are determined by using cross validation, and the determined prior distribution is applied to the context clustering. The results on continuous phoneme recognition experiments demonstrated that the proposed method outperformed the context clustering based on the MDL criterion and cross validation with ML estimates. The proposed method could determine prior distributions without any tuning parameters, and select the model structure which can accurately predict acoustic features for each HMM state. As future work, we will apply a Bayesian criterion using cross validation for selecting the number of mixtures, and apply a Bayesian criterion which represents the classification performance directly to the context clustering. In Chapter 4, the new framework of speech synthesis based on the Bayesian approach was proposed. In the proposed framework, all processes for constructing the system could be derived from one single predictive distribution which represents the problem of speech synthesis directly. The results on the MOS test demonstrated that the proposed method outperform the conven-

tional one. Our future work will include investigation of the relation between the speech quality and the size of model structure. And also, a speech synthesis technique integrating training and synthesis processes based on the Bayesian framework was proposed in Chapter 4. The proposed method removed the approximation that the posterior distribution of the model parameters is independent of the synthesis data and derived an algorithm that the posterior distributions, decision trees and synthesis data are iteratively updated. Both sentence-form and batch-form integrations were tested. The sentence-form integration estimates different posterior distributions and decision trees for each synthesis sentence, whereas the batch-form integration estimates the same ones for all synthesis sentences. The results of MOS synthesis demonstrated that the proposed method outperforms the baseline method and the sentence-form integration performed better than the batch-form integration. Future work will include investigation of the relation between the amount of training data and the quality of speech synthesized by the proposed method. Chapter 5 has provided an analysis of the impacts of machine translation and speech synthesis on speech-to-speech translation. It has been shown that the naturalness and intelligibility of the synthesized speech are strongly affected by the fluency of the translated sentences. The intelligibility of synthesized speech is improved as the translated sentence become more fluent. In addition, it was found that long-span word N-gram scores correlate well with the perceived fluency of sentences and that phoneme N-gram scores correlate well with the perceived naturalness of synthesized speech. Future work will include investigations into the integration of machine translation and speech synthesis using word N-gram and phoneme N-gram scores.

Bibliography

[1] P. Koehn, F.J. Och, and D. Marcu. Statistical phrase-based translation. *Proceedings of NAACL-HLT 2003*, pp. 127–133, 2003.

[2] F.J. Och and H. Ney. The alignment template approach to statistical machine translation. *Computational Linguistics*, Vol. 30, No. 4, pp. 417–449, 2004.

[3] D. Wu. Stochastic inversion transduction grammars, with application to segmentation, bracketing, and alignment of parallel corpora. *Proceedings of IJCAI 1995*, pp. 1328–1334, 1995.

[4] D. Wu. Stochastic inversion transduction grammars and bilingual parsing of parallel corpora. *Computational Linguiatics*, pp. 377–403, 1997.

[5] H. Yamamoto, H. Okuma, and E. Sumita. Imposing constraints from the source tree on ITG constraints for SMT. *Proceedings of ACL Workshop SSST-2*, pp. 1–9, 2008.

[6] H.D. Huang, Y. Ariki, and M.A. Jack. Hidden Markov models for speech recognition. *EDINBURGH UNIVERSITY*, pp. 119–125, 1990.

[7] A. Gelman, J.B. Carlin, H.S. Stern, and D.B. Rubin. Bayesian data analysis. *Chapman & Hall*, 1995.

[8] H. Jiang, K. Hirose, and Q. Huo. Robust speech recognition based on a Bayesian prediction approach. *IEEE Transactions on Speech and Audio Processing*, Vol. 7, pp. 426–440, 1999.

[9] Q. Huo and C.H. Lee. A Bayesian predictive classification approach to robust speech recognition. *IEEE Transactions on Speech and Audio Processing*, Vol. 8, pp. 200–204, 2000.

[10] S. Watanabe and A. Nakamura. Effects of Bayesian predictive classification using variational Bayesian posteriors for sparse training data in speech recognition. *Proceedings of Interspeech 2005*, pp. 1105–1108, 2005.

[11] S. Watanabe, Y. Minami, A. Nakamura, and N. Ueda. Variational Bayesian estimation and clustering for speech recognition. *IEEE Transactions on Speech and Audio Processing*, Vol. 12, pp. 365–381, 2004.

[12] A. Watanabe, S.and Sako and A. Nakamura. Automatic determination of acoustic model topology using variational Bayesian estimation and clustering. *Proceedings of ICASSP 2004*, Vol. 1, pp. 813–816, 2004.

[13] R. Kohavi. A study of cross-valication and bootstrap for accuracy estimation and model selection. *Proceedings of International Joint Conference on AI*, pp. 1137–1145, 1995.

[14] S. Arlot and A. Celisse. A survey of cross-valication procedures for model selection. *Statistics Surveys*, Vol. 4, pp. 40–79, 2010.

[15] T. Shinozaki. HMM state clustering based on efficient cross-validation. *Proceedings of ICASSP*, Vol. 1, pp. 1157–1160, 2006.

[16] T. Yoshimura, K. Tokuda, T. Masuko, T. Kobayashi, and T. Kitamura. Simultaneous modeling of spectrum, pitch and duration in HMM-based speech synthesis. *Proceedings of Eurospeech 1999*, pp. 2347–2350, 1999.

[17] H. Attias. Inferring parameters and structure of latent variable models by variational Bayes. *Proceedings of UAI 15*, 1999.

[18] K. Hashimoto, H. Zen, Y. Nankaku, T. Masuko, and K. Tokuda. A Bayesian approach to HMM-based speech synthesis. *Proceedings of ICASSP 2009*, pp. 4029–4032, 2009.

[19] K. Hashimoto, Y. Nankaku, and K. Tokuda. A Bayesian approach to hidden semi Markov model based speech synthesis. *Proceedings of Interspeech 2009*, pp. 1751–1754, 2009.

[20] E. Vidal. Finite-State Speech-to-Speech Translation. *Proceedings of ICASSP 1997*, pp. 111–114, 1997.

[21] H. Ney. Speech translation: coupling of recognition and translation. *Proceedings of ICASSP 1999*, pp. 1149–1152, 1999.

[22] Q. Quirk, A. Menezes, and C. Cherry. Dependency treelet translation: Syntactically informed phrasal SMT. *Proceedings of ACL 2005*, pp. 271–279, 2005.

[23] L. Huang, K. Knight, and A. Joshi. Statistical syntax-directed translation with extended domain of locality. *Proceedings of AMTA 2006*, 2006.

[24] K. Yamada and K. Knight. A syntax-based statistical translation model. *Proceedings of ACL 2000*, pp. 523–530, 2000.

[25] D. Marcu, W. Wang, A. Echihabi, and K. Knight. SPMT: statistical machine translation with syntactified target language phrases. *Proceedings of EMNLP 2006*, pp. 44–52, 2006.

[26] D. Melamed. Statistical machine translation by parsing. *Proceedings of ACL 2004*, pp. 653–660, 2004.

[27] A.L. Berger, P.F. Brown, S.A.D. Pietra, V.J.D. Pietra, A.S. Kehler, and R.L. Mercer. Language translation apparatus and method of using context-based translation models. *United States patent, patent number 5510981*, 1996.

[28] C. Tillmann. A unigram orientation model for statistical machine translation. *Proceedings of NAACL-HLT 2004*, pp. 101–104, 2004.

[29] F.J. Och and H. Ney. A systematic comparison of various statistical alignment models. *Computational Linguistics*, Vol. 29, No. 1, pp. 19–51, 2003.

[30] H.J. Fox. Phrasal cohesion and statistical machine translation. *Proceedings of EMNLP 2002*, pp. 304–311, 2002.

[31] Japanese-English paper abstruct corpus. http://www.jst.go.jp.

[32] M. Uchiyama and H. Isahara. A Japanese-English patent parallel corpus. *Proceedings of AMTA MT summit XI*, pp. 475–482, 2007.

[33] A. Stolcke. SRILM – An extensible language model toolkit. *Proceedings of ICSLP 2002*, pp. 901–904, 2002.

[34] R. Kneser and H. Ney. Improved backing-off for m-gram language model. *Proceedings of ICASSP 1995*, pp. 181–184, 1995.

[35] F.J. Och. Minimum error rate training for statistical machine translation. *Proceedings of ACL 2003*, pp. 160–167, 2003.

[36] K. Papineni, S. Roukos, T. Ward, and W.J. Zhu. Bleu: a method for automatic evaluation of machine translation. *Proceedings of ACL 2002*, pp. 311–318, 2002.

[37] E. Charniak. A maximum-entropy-inspired parser. *Proceedings of NAACL 2000*, pp. 132–139, 2000.

[38] Chasen. http://chasen-legacy.sourceforge.jp/.

[39] Moses. http://www.statmt.org/moses/.

[40] M. Galley, M. Hopkins, K. Knight, and D. Marcu. What's in a translation rule? *Proceedings of NAACL-HTL 2004*, 2004.

[41] K.F. Lee. Context-dependent phonetic hidden Markov models for speaker-independent continuous speech recognition. *IEEE Transactions on Acoustics, Speech and Signal Processing*, Vol. 38, No. 4, pp. 599–609, 1990.

[42] J.J. Odell. The use of context in large vocabulary speech recognition. *Cambridge University*, 1995.

[43] S. Young, J.J. Odell, and P. Woodland. Tree-based state tying for high accuracy acoustic modelling. *Proceedings of ARPA Workshop on Human Language Technology*, pp. 307–312, 1994.

[44] K. Shinoda and T. Watanabe. Acoustic modeling based on the MDL criterion for speech recognition. *Proceedings of Eurospeech 1997*, pp. 99–102, 1997.

[45] G. Schwarz. Estimating the dimension of a model. *The annals of staistics*, Vol. 6, No. 2, pp. 461–464, 1978.

[46] H. Robbins. An empirical Bayes approach to statistics. *Proceedings of 3rd Berkeley Symposium on Mathematical Statistics and Probability*, Vol. 1, pp. 157–163, 1956.

[47] A.P. Dempster, N.M. Laird, and D.B. Rubin. Maximum-likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society Series B (Methodological)*, Vol. 39, No. 1, pp. 1–38, 1977.

[48] 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*, Vol. 2, pp. 291–298, 1994.

[49] S. Kullback and R.A. Leibler. On information and sufficiency. *Annals of Mathematical Statistics*, Vol. 22, pp. 79–86, 1951.

[50] 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. *Journal of the Acoustical Society of Japan (E)*, Vol. 20, No. 3, pp. 199–206, 1999.

[51] K. Tokuda, T. Yoshimura, T. Masuko, T. Kobayashi, and T. Kitamura. Speech parameter generation algorithms for HMM-based speech synthesis. *Proceedings of ICASSP 2000*, pp. 936–939, 2000.

[52] H. Zen, K. Tokuda, T. Masuko, T. Kobayashi, and T. Kitamura. Hidden semi-Markov model based speech synthesis. *Proceedings of ICSLP*, pp. 1185–1180, 2004.

[53] A. Kurematsu, K. Takeda, Y. Sagisaka, S. Katagiri, H. Kuwabara, and K. Shikano. ATR Japanese speech database as a tool of speech recognition and synthesis. *Speech Communication*, Vol. 9, pp. 357–363, 1990.

[54] K. Tokuda, T. Masuko, N. Miyazaki, and T. Kobayashi. Hidden Markov models based on multi-space probability distribution for pitch pattern modeling. *Proceedings of ICASSP 1999*, pp. 229–232, 1999.

[55] The emime project. http://www.emime.org/.

[56] Y.J. Wu, Y. Nankaku, and K. Tokuda. State mapping based method for cross-lingual speaker adaptation in HMM-based speech synthesis. *Proceedings of Interspeech 2009*, pp. 528–531, 2009.

[57] M. Bulyko, I.and Ostendorf. Efficient integrated response generation from multiple target using weighted finite state transducers. *Computer Speech and Language*, Vol. 16, pp. 533–550, 2002.

[58] C. Nakatsu and M. While. Learning to say it well: Reranking realizations by predicted synthesis quality. *Proceedings of ACL 2006*, 2006.

[59] C. Boidin, V. Rieser, L.V.D. Plas, O. Lemon, and J. Chevelu. Predicting how it sounds: Re-ranking dialogue prompts based on TTS quality for adaptive spoken dialogue systems. *Proceedings of Interspeech 2009*, pp. 2487–2490, 2009.

[60] Amazon mechanical turk. https://www.mturk.com/.

[61] A.D. Gispert, S. Virpioja, M. Kurimo, and W. Byrne. Minimum Bayes risk combination of translation hypotheses from alternative morphological decompositions. *Proceedings of NAACL-HLT 2009*, pp. 73–76, 2009.

[62] G. Iglesias, A.D. Gispert, E.R. Banga, and W. Byrne. Hierarchical phrase-based translation with weighted finite state transducers. *Proceedings of NAACL-HLT 2009*, pp. 433–441, 2009.

[63] P. Koehn. Europarl: A parallel corpus for statistical machine translation. *Proceedings of MT Summit*, pp. 79–86, 2005.

[64] Hmm-based speech synthesis system (hts). http://hts.sp.nitech.ac.jp/.

[65] H. Kawahara, I. Masuda-Katsuse, and A. Cheveigne. Restructuring speech representations using a pitch-adaptive time-frequency smoothing and an instantaneous-frequency-based F0 extraction: Possible role of a repetitive structure in sounds. *Speech Communication*, Vol. 27, pp. 187–207, 1999.

[66] Festival. http://www.festvox.org/festival/.

[67] J.S. White, T. O'Connell, and F. O'Mara. The ARPA MT evaluation methodologies: evolution, lessons, and future approaches. *Proceedings of AMTA*, pp. 193–205, 1994.

[68] L. Malfait, J. Berger, and M. Kastner. P.563 – The ITU-T standard for signal-ended speech quality assesment. *IEEE Transactions on Audio, Speech and Language Processing*, Vol. 14, No. 6, pp. 1924–1934, 2006.

List of Publications

Journal papers

[**1**] **Kei Hashimoto**, Hirohumi Yamamoto, Hideo Okuma, Eiichiro Sumita, and Keiichi Tokuda, "A reordering model using a source-side parse-tree for statistical machine translation," *IEICE Transactions on Information & Systems*, vol. E92-D, no. 12, pp. 2386–2393, Dec. 2009.

[**2**] **Kei Hashimoto**, Heiga Zen, Yoshihiko Nankaku, Akinobu Lee, and Keiichi Tokuda, "Bayesian context clustering using cross validation for speech recognition," *IEICE Transactions on Information & Systems*, (accepted).

[**3**] Sayaka Shiota, **Kei Hashimoto**, Heiga Zen, Yoshihiko Nankaku, Akinobu Lee, and Keiichi Tokuda, "Speech recognition based on statistical models including multiple model structures," *Acoustical Science and Technology*. (submitted)

International conference proceedings

[**3**] **Kei Hashimoto**, Heiga Zen, Yoshihiko Nankaku, Akinobu Lee, and Keiichi Tokuda, "Hyperparameter estimation for speech recognition based on variational Bayesian approach," *Proceedings of ASA & ASJ Joint Meeting*, p. 3042, Dec. 2006.

[**4**] **Kei Hashimoto**, Heiga Zen, Yoshihiko Nankaku, Akinobu Lee, and Keiichi Tokuda, "Bayesian context clustering using cross valid prior distribution for HMM-based speech recognition," *Proceedings of Interspeech 2008*, pp. 936–939, Sep. 2008.

[**5**] Sayaka Shiota, **Kei Hashimoto**, Heiga Zen, Yoshihiko Nankaku, Akinobu Lee, and Keiichi Tokuda, "Acoustic modeling based on model structure annealing for speech recognition," *Proceedings of Interspeech 2008*, pp. 932–935, Sep. 2008.

[**6**] Tatusya Ito, **Kei Hashimoto**, Heiga Zen, Yoshihiko Nankaku, Akinobu Lee, and Keiichi Tokuda, "Speaker recognition based on variational Bayesian method," *Proceedings of Interspeech 2008*, pp. 1417–1420, Sep. 2008.

[**7**] **Kei Hashimoto**, Heiga Zen, Yoshihiko Nankaku, Takashi Masuko, and Keiichi Tokuda, "A Bayesian approach to HMM-based speech synthesis," *Proceedings of ICASSP 2009*, pp. 4029–4032, April 2009.

[**8**] **Kei Hashimoto**, Hirohumi Yamamoto, Hideo Okuma, Eiichiro Sumita, and Keiichi Tokuda, "Reordering model using syntactic information of a source tree for statistical machine translation," *Proceedings of NAACL-HLT 2009 Workshop SST-3*, pp. 69–77, June 2009.

[**9**] **Kei Hashimoto**, Yoshihiko Nankaku, and Keiichi Tokuda, "A Bayesian approach to hidden semi Markov model based speech synthesis," *Proceedings of Interspeech 2009*, pp. 1751–1754, Sep. 2009.

[**10**] Sayaka Shiota, **Kei Hashimoto**, Yoshihiko Nankaku, and Keiichi Tokuda, "Deterministic annealing based training algorithm for Bayesian speech recognition," *Proceedings of Interspeech 2009*, pp. 680–683, Sep. 2009.

[**11**] **Kei Hashimoto**, Yoshihiko Nankaku, and Keiichi Tokuda, "Bayesian speech synthesis framework integrating training and synthesis processes," *Proceedings of SSW7*, pp. 106–111, Sep. 2010.

[**12**] Keiichiro Oura, **Kei Hashimoto**, Sayaka Shiota, and Keiichi Tokuda, "Overview of NIT HMM-based speech synthesis system for Blizzard Challenge 2010," *Proceedings of Blizzard Challenge 2010*, Sep. 2010.

[**13**] **Kei Hashimoto**, Junichi Yamagishi, William Byrne, Simon King, and Keiichi Tokuda, "An analysis of machine translation and speech synthesis for speech-to-speech translation," *Proceedings of ICASSP 2011*. (submitted)

Technical reports

[**13**] Sayaka Shiota, **Kei Hashimoto**, Heiga Zen, Yoshihiko Nankaku, Akinobu Lee, and Keiichi Tokuda, "Speech recognition based on statistical models including multiple decision trees," *Technical Report of IEICE*, vol. 108, no. 338, pp. 221–226, Dec. 2008.

[**14**] **Kei Hashimoto**, Heiga Zen, Yoshihiko Nankaku, and Keiichi Tokuda, "Acoustic Modeling Based on Model Structure Annealing for Speech Recognition," *Technical Report of IEICE*, vol. 107, no. 165, pp. 67–72, Dec. 2008.

[**15**] Sayaka Shiota, **Kei Hashimoto**, Heiga Zen, Yoshihiko Nankaku, Akinobu Lee, and Keiichi Tokuda, "Speech recognition based on statistical models including multiple decision trees," *Technical Report of IEICE*, vol. 108, no. 338, pp. 221–226, Dec. 2008.

[**16**] Tatsuya Ito, **Kei Hashimoto**, Heiga Zen, Yoshihiko Nankaku, Akinobu Lee, and Keiichi Tokuda, "Speaker recognition based on Gaussian mixture models using variational Bayesian method," *Technical Report of IEICE*, vol. 108, no. 338, pp. 185–190, Dec. 2008.

Domestic conference proceedings

[**17**] **Kei Hashimoto**, Heiga Zen, Yoshihiko Nankaku, Akinobu Lee, and Keiichi Tokuda, "Hyper-parameter tying structure for speech recognition based on variational Bayesian method," *Proceedings of Autumn Meeting of the ASJ*, pp. 139–142, Sep. 2007.

[**18**] Sayaka Shiota, **Kei Hashimoto**, Heiga Zen, Yoshihiko Nankaku, Akinobu Lee, and

Keiichi Tokuda, "Acoustic modeling based on model structure annealing for speech recognition," *Proceedings of Autumn Meeting of the ASJ*, pp. 143–146, Sep. 2007.

[**19**] **Kei Hashimoto**, Heiga Zen, Yoshihiko Nankaku, Akinobu Lee, and Keiichi Tokuda, "Context clustering using cross validation based on Bayesian criterion," *Proceedings of April Meeting of the ASJ*, pp. 69–70, Mar. 2008.

[**20**] Tatsuya Ito, **Kei Hashimoto**, Heiga Zen, Yoshihiko Nankaku, Akinobu Lee, and Keiichi Tokuda, "Speaker recognition based on variational Bayesian method," *Proceedings of April Meeting of the ASJ*, pp. 143–144, Mar. 2008.

[**21**] **Kei Hashimoto**, Heiga Zen, Yoshihiko Nankaku, Akinobu Lee, and Keiichi Tokuda, "HMM based speech synthesis using cross validation for Bayesian criterion," *Proceedings of Autumn Meeting of the ASJ*, pp. 251–252, Sep. 2008.

[**22**] Sayaka Shiota, **Kei Hashimoto**, Heiga Zen, Yoshihiko Nankaku, Akinobu Lee, and Keiichi Tokuda, "Speech recognition based on multiple phonetic decision tree structures," *Proceedings of Autumn Meeting of the ASJ*, pp. 125–126, Sep. 2008.

[**23**] **Kei Hashimoto**, Yoshihiko Nankaku, and Keiichi Tokuda, "Hidden semi-Markov model based Bayesian speech synthesis," *Proceedings of April Meeting of the ASJ*, pp. 303–304, Mar. 2009.

[**24**] **Kei Hashimoto**, Yoshihiko Nankaku, and Keiichi Tokuda, "Evaluation of HSMM-based speech synthesis based on Bayesian framework," *Proceedings of Autumn Meeting of the ASJ*, pp. 257–258, Sep. 2009.

[**25**] Sayaka Shiota, **Kei Hashimoto**, Yoshihiko Nankaku, and Keiichi Tokuda, "Training algorithm based on deterministic annealing for Bayesian speech recognition," *Proceedings of Autumn Meeting of the ASJ*, pp. 3–6, Sep. 2009.

[**26**] **Kei Hashimoto**, Yoshihiko Nankaku, and Keiichi Tokuda, "Bayesian speech synthesis integrating training and synthesis processes," *Proceedings of Autumn Meeting of the ASJ*, pp. 243–244, Sep. 2010.

Appendix A

Samples from the English to Japanese Translation

Sample 1

Source: The free-running period (τ) in constant darkness (DD) at 20 ℃ became shorter than that at 25 ℃, suggesting that the phase advance of locomotor activity in LD cycles at 20 ℃ was caused by the decrease in τ .

Baseline: 一定の暗闇で２０℃で２５℃でのそれに比べ短くなっていることから，２０℃でＬＤ周期で歩行活動の位相前進（τ）のフリーランニング期間（ＤＤ）の減少が原因であった。

Reference: ２０℃での一定の暗環境（ＤＤ）における自由継続期（τ）は２５℃よりも短くなり，２０℃でのＬＤ周期における運動活動の位相前進がτの減少に起因することを示唆した。

IST-ITG: ２０℃で一定の暗闇（ＤＤ）でフリーランニング期間（τ）よりも短くなり，２５℃で２０℃でＬＤ周期で歩行活動の進相であることが示唆され，τの減少が原因であった。

Proposed: ２０℃で一定の暗闇（ＤＤ）でフリーランニング期間（τ）２５℃でのそれに比べ短くなっていることから，２０℃でＬＤ周期で歩行活動の位相前進τの減少が原因であった。

Sample 2

Source: From result of the consideration, it was pointed that radiation from the loop elements was weak.
Baseline: 考察の結果からことを指摘し，ループ素子からの放射は弱かった。
Reference: 考察結果より，ループ素子からの放射が弱いことを指摘する。
IST-ITG: 考察の結果から，ことを指摘し，ループの要素からの放射は弱かった。
Proposed: 考察の結果から，ループ素子からの放射は弱いことを示した。

Sample 3

Source: The value of TF, on the other hand, was higher in the reverse order, indicating that high oxidation rate causes severe defects on the surface of Ni crystallites.
Baseline: 一方，重症の表面上の欠陥の原因となることを示し，逆順に高かったが，ＴＦの値は高い酸化速度はＮｉの微結晶た。
Reference: 一方，ＴＦの値は逆の順序で高く，酸化速度が高いことはＮｉ結晶の表面欠陥の原因になることを示した。
IST-ITG: 一方，ＴＦの値は逆の順序で高かったことを示し，高い酸化速度は表面にＮｉ微結晶の重篤な欠陥の原因となる。
Proposed: 一方，ＴＦの値は逆の順序で高かったことを示し，高い酸化速度は，Ｎｉの微結晶表面に重篤な欠陥の原因となる。

Sample 4

Source: Still, as well as the preceding fiscal year, local area explanatory meetings of the construction, noise reductions, and so on, were opened on suitable occasions, for neighbourhood inhabitant of Shinjuku ward and Shibuya ward.
Baseline: なお，近隣住民説明会，局所領域は前年度と同様の建設，低騒音化，などを開設した新宿区のために，渋谷区及びケースのに適している。
Reference: なお，前年度と同様に，工事の節目に新宿区，渋谷区の近隣住民を対象に地元説明会を行い低騒音化等について説明した。
IST-ITG: なお，前年度と同様，低雑音，地元説明会の構築のため，渋谷区及び新宿区の近隣住民のために，適切なケース上に公開した。
Proposed: なお，前年度と同様，低雑音などの施工，地元説明会，渋谷区及び新宿区の近隣住民を対象にケースを適当に開設された。

Sample 5

Source: Aggravation was obvious from the latter half of March to the end of April, and he contracted the disease in February to the beginning of May.

Baseline: ４月末に３月後半から５月上旬に２月に疾患を発症し，著明な増悪した。

Reference: ３月後半から４月末に増悪が著明で，２～５月上旬に発症した。

IST-ITG: ３月後半から４月末に著明な増悪し，５月上旬に２月に疾患を発症した。

Proposed: ３月後半から４月末に著明な増悪し，５月上旬に２月に疾患を発症した。

Sample 6

Source: After diagnosing the pleural effusion and ascites, vein catheter was left in place under the echo guide, and after removing the pleural effusion and ascites, OK-432 was administered locally.

Baseline: 診断後，胸水，腹水，胸水・腹水を除去した後，エコーガイド下で，静脈カテーテルを左に代わってＯＫ‐４３２を投与した。

Reference: 胸水・腹水の診断を行った後にエコーガイド下に静脈カテーテルを留置し，胸水・腹水を除去し，ＯＫ‐４３２を局所投与した。

IST-ITG: 胸水・腹水の診断後，静脈カテーテルを残したエコーガイド下で代わりに，胸水・腹水を除去した後，ＯＫ‐４３２，局所的に投与した。

Proposed: 胸水・腹水の診断後，静脈カテーテルを残したエコーガイド下で代わりに，胸水・腹水を除去した後，ＯＫ‐４３２，局所的に投与した。

Appendix B

Software

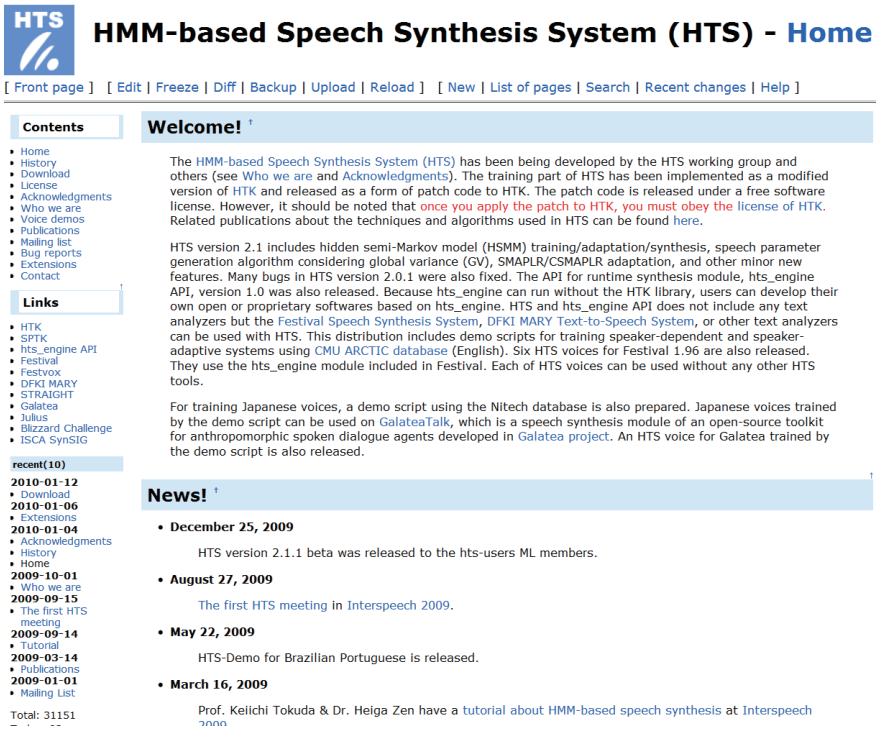

Figure B.1: HTS: http://hts.sp.nitech.ac.jp/