Building Speech Recognition System from Untranscribed Data

Report from JHU workshop 2016

Lukáš Burget [1], Sanjeev Khudanpur [2], Najim Dehak [2], Jan Trmal [2], Reinhold Haeb-Umbach [3], Graham Neubig [4], Shinji Watanabe [5], Daichi Mochihashi [6], Takahiro Shinozaki [7], Ming Sun [8], Chunxi Liu [2], Matthew Wiesner [2], Raghavendra Pappagari [2], Lucas Ondel [1], Mirko Hannemann [1], Santosh Kesiraju [9,1], Thomas Glarner [3], Leda Sari [10], Jinyi Yang [11], Ondřej Cífka [12], Yibo Yang [13], Alena Rott [14], and Jan "Honza" Černocký (editor of the report) [1]

(1) Brno University of Technology, Czech Republic,
(2) Johns Hopkins University, USA,
(3) University of Paderborn, Germany
(4) Nara Institute of Science and Technology, Japan
(5) Mitsubishi Electric Research Laboratory, USA
(6) Institute of Statistical Mathematics, Japan
(7) Tokyo Institute of Technology, Japan
(8) Amazon, USA
(9) International Institute of Information Technology, Hyderabad, India
(10) University of Illinois, USA
(11) University of Chinese Academy of Sciences, China
(12) Charles University in Prague, Czech Republic
(13) University of Texas at Dallas, USA
(14) Stanford University, USA

Team members

Team Leader

Lukáš Burget	Brno University of Technology

Senior Researchers

Sanjeev Khudanpur	Johns Hopkins University
Najim Dehak	Johns Hopkins University
Jan Trmal	Johns Hopkins University
Reinhold Haeb-Umbach	University of Paderborn
Graham Neubig	Nara Institute of Science and Technology
Shinji Watanabe	Mitsubishi Electric Research Laboratory
Daichi Mochihashi	Institute of Statistical Mathematics
Takahiro Shinozaki	Tokyo Institute of Technology
Ming Sun	Amazon

Graduate Students

Chunxi Liu	Johns Hopkins University
Matthew Wiesner	Johns Hopkins University
Raghavendra Pappagari	Johns Hopkins University
Lucas Ondel	Brno University of Technology
Mirko Hannemann	Brno University of Technology
Santosh Kesiraju	IIIT Hyderabad
Thomas Glarner	University of Paderborn
Leda Sari	University of Illinois
Jinyi Yang	University of Chinese Academy of Sciences

Undergraduate Students

Ondřej Cífka	Charles University in Prague
Yibo Yang	University of Texas at Dallas
Alena Rott	Stanford University

Acknowledgements

The work reported here was carried out during the 2016 Jelinek Memorial Summer Workshop on Speech and Language Technologies, which was supported by Johns Hopkins University via DARPA LORELEI Contract No HR0011-15-2-0027, and gifts from Microsoft, Amazon, Google, Facebook. Shinji Watanabe was supported by JSPS KAKENHI Grant Number 26280055 and Mitsubishi Electric Research Laboratories (MERL).

Contents

Chapter 1

Introduction

1.1 Planned work

1.1.1 Introduction

Modern automatic speech recognition (ASR) systems consist of two major statistical models: the Language Model (LM) and the Acoustic Model (AM). The LM models probabilities of word sequences and the AM describes distributions of acoustic features for individual phones (or senones). Typically, the two statistical models are independently trained from large volumes of text data and annotated speech data, respectively. The component connecting these two models is the pronunciation lexicon mapping words into phone sequences. The pronunciation lexicon is typically manually designed by an expert familiar with the language of interest. Recently, there has been an increased interest (e.g. in IARPA Babel and DARPA LORELEI programs) in rapidly developing ASR systems for new "exotic" low-resource languages, where such expert-level linguistic input and manual speech transcription are too expensive, too time consuming, or simply impossible to obtain.

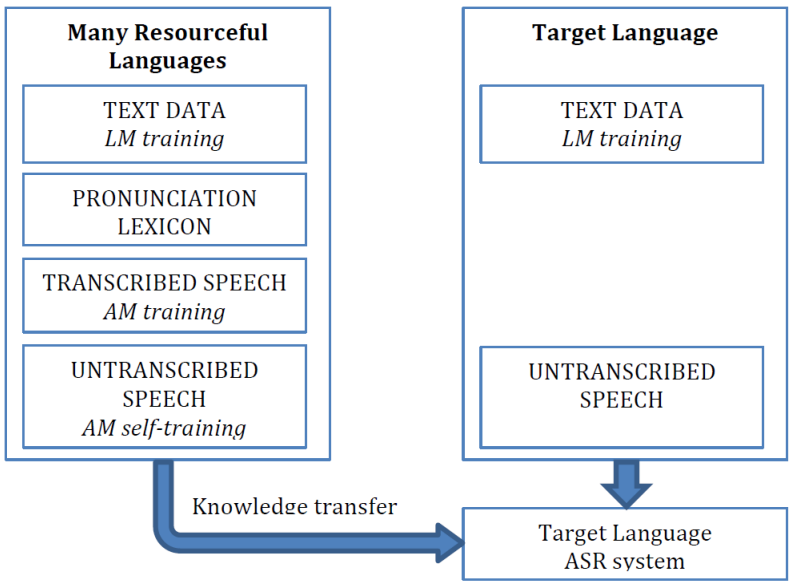


Figure 1.1: Overall scheme of the work planned.

As illustrated in Figure 1.1, we propose to develop models and techniques that will allow us to train an ASR system for a new low-resource target language, where only text data and "unrelated" untranscribed speech recordings are available. During the training, the proposed models must be able to reveal and match the patterns seen in text data (i.e. the regularities seen in the word sequences)

with similar patterns observed in speech signal. To accomplish this task reliably, we propose to leverage data from other high-resource languages, for which transcribed data and expert knowledge are available. For example, to discover patterns of phone-like units in speech, it is important to discriminate between the phonetic variability in the speech signal and the variability attributed to other causes (speaker, channel, noise, etc.). To a large extent, this knowledge can be learned from the transcribed speech of the high-resource languages and used for building the target ASR system.

1.1.2 Research

Recently, discriminatively trained Deep Neural Networks (DNNs) have been very successful in ASR and have largely superseded the more traditional generative models. We also plan to use DNNs to facilitate knowledge transfer from the high-resource languages as will be described later. However, this project mainly focuses on Bayesian generative models, which are more suitable for the unsupervised discovery of latent patterns in untranscribed data. As illustrated in Figure 1.2, the envisioned generative model consists of the same components as the traditional model for ASR. A sequence of words is assumed to be generated from a "known" statistical language model, which can be estimated from the available text data. The word (or corresponding letter) sequence is converted into a sequence of acoustic (phone-like) units using the lexicon model. Finally, the corresponding sequence of observed speech features is assumed to be generated from the acoustic model. However, unlike in the case of the traditional ASR system with a handcrafted lexicon and supervised acoustic model training, a proper Bayesian non-parametric model will be used to represent the lexicon and acoustic model in order to jointly solve the following problems arising during the unsupervised training:

1. dividing speech into phone-like segments,
2. clustering the segments to obtain a repository of acoustic phone-like units
3. learning the corresponding acoustic model (i.e. acoustic unit feature distributions)
4. learning the lexicon as a statistical model for translating letter sequence into a sequence of the discovered (possibly context-dependent) acoustic units, and
5. discovering sequences of acoustic units and words that are in agreement with the language model.

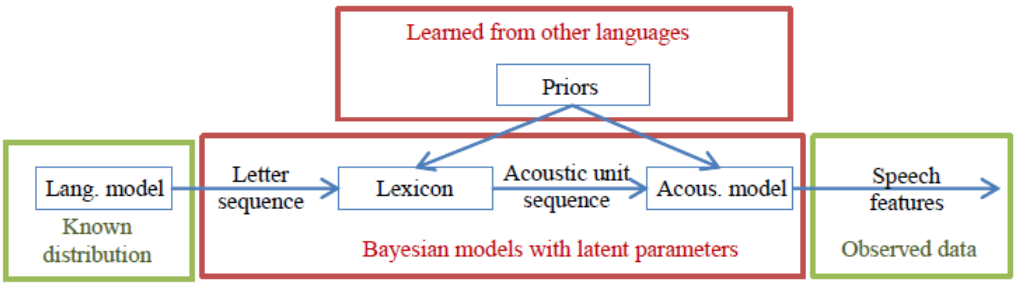


Figure 1.2: Envisioned generative model.

The important and novel feature of our model will be the possibility to learn from the high-resource languages. In the framework of Bayesian generative models, we can design a model where some of the parameters and latent variables are shared across languages. In other words, we can assume that speech from all languages is generated from a single properly defined generative model. Some of the variables can be considered as observed for the supervised languages and hidden for the target low-resource language. Posterior distributions over some latent variables estimated from the high-resource languages can be used as priors in the inference for the low-resource language.

1.1.3 Related Work

To give a more concrete idea of our envisioned model, we now review several previously proposed models, each focusing on some part of our problem. During the workshop, we would like to adapt these models to our needs and use them as the building blocks to solve the whole problem of training from untranscribed data. Note that the potential workshop participants are often the authors of the reviewed models or people with the appropriate expertise.

In [6], a Bayesian non-parametric model for acoustic unit discovery was proposed, based on a Dirichlet Process Mixture of HMMs. This model jointly solved the problem of 1) discovering acoustic units in speech signal, 2) segmenting speech into such units, and 3) learning their HMM models. In [50], the model was further extended with a pronunciation lexicon component based on Hierarchical Dirichlet distribution model. This model allowed the acoustic model to be trained from orthographically transcribed speech without any need for a handcrafted lexicon or definition of phonetic units by an expert. In these works, Gibbs sampling was used for inference, which made the training slow and impractical for application to larger data sets. Variational Bayesian inference is proposed to train a similar acoustic unit discovery model in [7], where improvements in both scalability and quality of the discovered acoustic units were reported. None of the models, however, made any attempt to leverage the data from the high-resource languages to improve the acoustic unit discovery for the target language.

One simple way of using the data from the high-resource languages, which we also plan to investigate, is to use DNN based multilingual bottle-neck (BN) speech features for training the acoustic unit discovery model. The BN features, which are discriminatively trained on multiple languages to suppress information irrelevant for phone discrimination, have already proved to provide excellent performance when building ASR systems for languages with limited amount of transcribed data [3].

The problem of acoustic unit discovery is very similar to speaker diarization. It was shown that a fully Bayesian model for speaker diarization can greatly benefit from Joint Factor Analysis inspired priors describing across-speaker variability in the space of speaker model parameters [2][1]. Similar priors describing within- and across-phone variabilities can be robustly trained from the high-resource languages and incorporated into our model for acoustic unit discovery. This concept is also similar to the Subspace GMM model, which was successfully used for multilingual acoustic modeling [1].

Non-parametric Bayesian models based on Hierarchical Pitman-Yor Processes were successfully used to learn language models over word-like units automatically discovered (using the same model) in phone sequences or lattices [54, 4]. These models, however, assume known acoustic units (phones), which can be obtained from continuous speech using an (error-free) phone recognizer. Heymann et al. extended this to a real (error-prone) phone recognizer and showed how, through alternating between language model learning and phone recognition, both improved phone error rates and improved word discovery F-scores could be obtained [124].

1.1.4 Expected Outcomes

We expect to develop a framework for training ASR systems from untranscribed speech applicable to data sets of non-trivial size. Also, there will be no need for costly and time consuming construction of a pronunciation dictionary by an expert linguist. This framework should allow for a speech representation that is flexible enough to integrate knowledge from existing languages but also independent enough to discover new patterns in an unsupervised way. To achieve this goal, we will develop nontrivial extensions to the aforementioned models and combine them into a single functioning framework. Meta-heuristic optimization [5] will be used to aid the search for the optimal model configuration and parameter settings.

The important part of our problem is the model for acoustic unit discovery, which will allow us to

[1] `http://speech.fit.vutbr.cz/software/vb-diarization-eigenvoice-and-hmm-priors`

convert speech into discrete high-quality phone-like units. Besides the ASR task, this model might be also useful for a range of other speech applications, where reliable tokenization is requested (speaker recognition, language identification, query-by-example keyword spotting, etc). Therefore, we also plan to evaluate the quality of the discovered acoustic unit sequences by

1. their direct comparison with the true phone sequences (e.g. using the normalized cross-entropy measure), as well as

2. testing their performance in some of the aforementioned speech applications.

As this is the first attempt to solve the whole problem of training ASR from untranscribed speech, we believe that the outcomes from the workshop will serve as a starting point for new research in these directions. Different models and approaches providing much better performance will certainly emerge soon.

1.2 Summary of work done and scope of chapters

The research performed during the workshop did not lead to the definition of a complete system, but made people from different disciplines work together and sparkled many ideas that have the potential to converge to the ultimate goal of building ASR from untranscribed data in the coming years. The work led to several collaborative papers, mainly for the forthcoming ICASSP 2017 conference in New Orleans — the following chapters mostly re-use the material from these submissions.

One of the main research avenues at the workshop was the automatic discovery of acoustic units (AUD). Chapter 2 covers Bayesian acoustic unit discovery with phonotactic language model and is actually an extension of Ondel's paper [7] that was the basis of significant amount of work at JHU. During the workshop, we have investigated into a non-parametric Bayesian phone-loop model where the prior over the probability of the phone-like units is assumed to be sampled from a Dirichlet Process (DP). The model is improved by incorporating a Hierarchical Pitman-Yor based bigram Language Model.

Chapter 3 investigates into alternative inference methods for AUD model training, particularly the collapsed Gibbs sampling and collapsed variational inference for finite Gaussian mixture models.

While the baseline approaches work with AUD in each language, we have also tried to improve the AUD performance in a resource-less language by transferring knowledge from other languages (Chapter 4). We present methods that either use the posterior estimates of the parameters in AUD models trained on different languages as the prior in the target language AUD training procedure or utilize information contained in multiple languages while extracting features of the target data for AUD training.

The workshop also generated a need to *evaluate the quality of AUD* without having the complete ASR system. Therefore, Chapter 5 investigates into three empirical strategies of AUD evaluation: (1) normalized mutual information (NMI) against orthographic phoneme transcripts, (2) same-different evaluation of word-pairs and (3) spoken document classification and clustering. Furthermore, Chapter 6 deals with assessing the quality of acoustic unit discovery on the task of topic identification of spoken documents. This is highly relevant for example for the ongoing DARPA Lorelei program, where the type of incidents need to be detected without a-priori knowledge of the target language.

Chapter 7 targets combining acoustic and lexical unit discovery towards complete unsupervised LVCSR. Sitting at the "core" of the proposed system, this chapter suggests a feedback from the word discovery to the acoustic model discovery unit to improve the latter by exploiting the language model information learned in the former.

Chapter 8 presents fully Bayesian grapheme-to-phoneme (G2P) conversion by joint-sequence models, improving over the traditional scheme defined by [62] and implemented in the popular Sequitur

tool[2]. The proposed scheme is fully transducer-based and combines well with other blocks necessary for the whole scheme.

The last block, producing human-readable words (as expected at the output of ASR), is addressed by Chapter 9 dealing with unsupervised learning of pronunciation dictionary from unaligned phone and word data. Although this task is very challenging and we are at the initial stage, we demonstrate that a model based on Bayesian learning of Dirichlet processes can acquire word pronunciations from phone transcripts and text.

Finally, Chapter 10 describes graphemic knowledge transfer. The assumption is that most low-resource languages are written using a phonemic orthography shared by a high-resource language and that furthermore, shared graphemes in these orthographies have similar acoustic realizations. We describe two methods of cross-lingual knowledge transfer by exploiting such shared orthographies.

As it is usual for JHU workshops, we expect a significant portion of new results with origins in hot Baltimore summer 2016 to appear in the forthcoming years and are looking forward to see the area of low- and zero-resource ASR flourishing.

[2] `https://www-i6.informatik.rwth-aachen.de/web/Software/g2p.html`

Chapter 2

Bayesian Acoustic Unit Discovery with Phonotactic Language Model

Lucas Ondel, Lukas Burget, Jan Cernocky, and Santosh Kesiraju

Recent work on Acoustic Unit Discovery (AUD) as led to the development of an non-parametric Bayesian phone-loop model where the prior over the probability of the phone-like units is assumed to be sampled from a Dirichlet Process (DP). In this work, we propose to improve this model by incorporating a Hierarchical Pitman-Yor based bigram Language Model on top of the units' transitions. This new model makes use of the phonotactic context information but assumes a fixed number of units. To remedy this limitation we first train a DP phone-loop model to infer the number of units, then, the bigram phone-loop is initialized from the DP phone-loop and trained until convergence of its parameters. Results show an absolute improvement of 1-2 % on the Normalized Mutual Information (NMI) metric. Furthermore, we show that, combined with Multilingual Bottleneck (MBN) features the model yields a same or higher NMI as an English phone recogniser trained on TIMIT.

2.1 Introduction

Whereas Automatic Speech Recognition (ASR) systems are more and more frequently used in daily life applications, the need for labeled data has never been so high. With the ever-growing use of Internet a huge amount of unlabeled audio data coming from many different countries is now available. However, because the labeling process by human expert is expensive this data has still been unexploited. In [6], a nonparametric Bayesian model that automatically segments and labels audio data has been proposed. The model was later refined in [7] in order to be trained using the Variational Bayes (VB) method. An attempt to tackle the problem by means of neural networks has also been investigated in [8]. In [7], the Acoustic Unit Discovery (AUD) is done by clustering temporal sequences with a Dirichlet Process (DP) based mixture model where, following the Variational treatment of the DP mixture model [9], the probability of the weights is approximated by a finite Categorical distribution. This distribution functions as a unigram Language Model (LM) over the units. This generative process is quite inaccurate as the probability of a phone (and by extension any phone-like unit) strongly depends on the previous phones. In the present work, we extend the AUD model described in [7] by replacing the naive Categorical distribution by a non-parametric Bayesian bigram LM. The chapter is organized as follows: Section 2.2 and 2.3 describes the original model and its extension respectively, Section 2.4 details the training of the extended model, Section 2.5 details how we evaluate the AUD task and finally, results are presented in Section 2.6.

2.2 Infinite Phone-Loop Model

Our model aims at segmenting and clustering unlabeled speech data into phone-like categories. It is similar to a phone-loop model in which each phone-like unit is modeled by an HMM [1]. This phone-loop model is fully Bayesian in the sense that:

- it incorporates a prior distribution over the parameters of the HMMs
- it has a prior distribution over the units modeled by a Dirichlet process [10].

Informally, the Dirichlet process prior can be seen as a standard Dirichlet distribution prior for a Bayesian mixture with an infinite number of components. However, we assume that our N data samples have been generated with only M components ($M \leq N$) from the infinite mixture. Hence, the model is no longer restricted to have a fixed number of components but instead can learn its complexity (i.e. number of components used M) according to the training data. The generation of a data set with M speech units can be summarized as follows:

1. sample the vector $\mathbf{v} = v_1, ..., v_M$ with

$$v_i \sim \text{Beta}(1, \gamma)$$

   where γ is the concentration parameters of the Dirichlet process

2. sample M HMM parameters θ_1,...,θ_M from the base distribution of the Dirichlet process

$$\theta_i \sim H$$

3. sample each segment as follows:

   (a) choose a HMM parameters with probability $\pi_i(\mathbf{v})$ defined as:

$$\pi_i(\mathbf{v}) = v_i \prod_{j=1}^{i-1}(1 - v_j)$$

   (b) sample a path $\mathbf{s} = s_1, ..., s_n$ from the HMM transition probability distribution

   (c) for each s_i in $\mathbf{s}$:

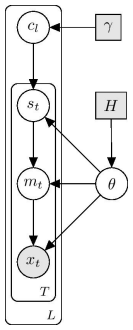
      i. choose a Gaussian component from the mixture model

      ii. sample a data point from the Gaussian density function

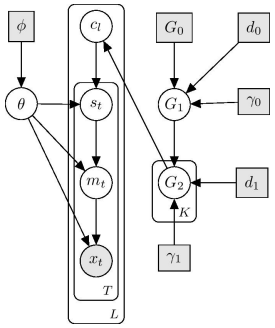
The graphical representation of this model is shown in Figure 2.1a. The priors over the GMM weights, Gaussian mean and (diagonal) covariance matrix are a Dirichlet and a Normal-Gamma density respectively. A similar model has been applied in [6], however, two major differences should be noted: first, we have chosen to consider the stick-breaking construction [9] of the Dirichlet process (step 1 and 2 of the generation) rather than the Chinese Restaurant Process (CRP). See [11] and [6] for training Bayesian models with the CRP. This allows us to use variational methods to infer the distribution over the parameters rather than sampling methods. Secondly, our model does not have any boundary variable. The segmentation of the data is carried out by seeing this mixture of HMMs as a single HMM and using the standard Viterbi algorithm. See [7] for the Variational Bayesian treatment of this model.

[1] By abuse of notation we write HMM for the complete HMM/GMM model.

Figure 2.1: Two different AUD models



(a) Phone-Loop model with a Dirichlet Process prior



(b) Phone-Loop model with a bigram phonotactic HPYLM

2.3 Bigram Phone-Loop Model

The model previously described is able to learn the appropriate number of units for a given data set thanks to the Dirichlet Process prior. The learnt probabilities of each unit to occur can be seen as a simple unigram phonotactic language model. It is well known however, that each language has a specific phone distribution and moreover a specific n-gram phone sequence distribution. Hence, the simple phone-loop model is limited in the sense that it does not make use of the phonotatic context information. To remedy this problem, we can replace the Dirichlet Process prior by a a Hierarchical Pitman-Yor process based Language Model (HPYLM) [12]. The HPYLM prior guarantees that the probability of each unit to occur depends on the previous O units, where O is the order of the hierarchy of the HPY. The data generation with a bigram based HPYLM is summarized as follows:

1. sample the HMM parameter sets $\theta_1, ..., \theta_K$ from the prior distribution:

$$\theta_i \sim \phi$$

2. sample a Categorical distribution from the top level Pitman-Yor process (PY)

$$G_1 \sim PY(G_0, \gamma_0, d_0)$$

where G_0, γ_0 and d_0 are the base distribution, the concentration and the discount parameters of the PY respectively. In our case, we assumed G_0 to be a uniform Categorical distribution

3. sample K context-dependent distributions over the units $G_{2,1}, ..., G_{2,K}$:

$$G_{2,i} \sim PY(G_1, \gamma_1, d_1)$$

where G_1, γ_1 and d_1 are the base distribution, the concentration and the discount parameters of the second-level PY respectively

4. sample each segment as follows:

   (a) sample the unit index for the c_t:

$$c_t \sim G_{2,c_{t-1}}$$

   (b) sample a sequence of features from the HMM with parameters θ_{c_t} as described in section 2.2

The graphical model corresponding to this generation process is depicted in Figure 2.1b. We draw the reader's attention to the fact that, contrary to the model presented in Section 2.2, we assume here a finite number of units. Hence, while the HPY based phone-loop can model context-dependent unit transitions, it is not suitable to infer the number of units. Eventually, this limitation could be resolved by assuming the HMM parameters $\boldsymbol{\theta}$ to be sampled from the top level base distribution G_0 of the HPY. However, because there is no known analytic form for the stick-breaking representation of the HPY [13], and therefore no simple VB inference algorithm adapted to this model, it would require Gibbs sampling to train the HMM parameters, losing the benefits of the VB inference, as discussed in [7].

2.4 Training

In section 2.2 and 2.3 we presented two phone-loop models, the first one learning the complexity (i.e. the number of units) needed to model the data whereas the latter one makes use of the phonotactic context information. Figure 2.2 shows the evolution of the number of units during the VB training of the DP based phone-loop model. As we can see, the number of units stabilizes very quickly at the beginning of the training. This suggests that we can proceed in two stages: first learning the number of units with the DP based phone-loop model and then refining the HMMs' parameters using the bigram phone-loop model. The DP phone-loop model is trained using VB inference as described in [7]. Once the training of the DP phone-loop model has converged we switch to a 3-steps training procedure that we repeat until convergence:

1. label the data with Viterbi algorithm using the current phone-loop model
2. train the HPY based language model on the labeled data using the Chinese Restaurant Franchise (CRF) [12]
3. set the unit-to-unit transitions according to the trained phonotactic LM and retrain the HMMs' parameters while keeping fixed the aforementioned transitions.

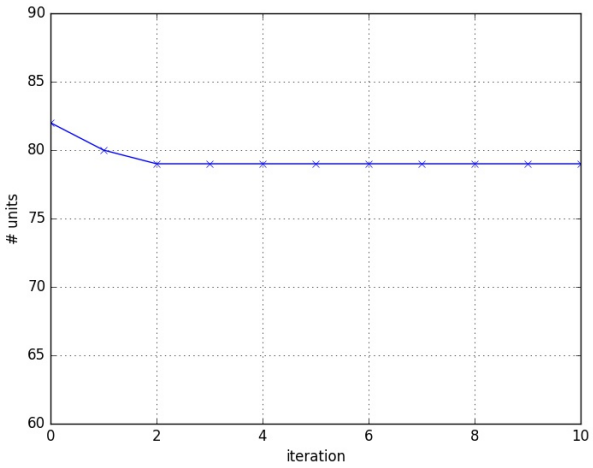


Figure 2.2: Evolution of the number of units during the training of the DP model

While this algorithm was experimentally proven to be efficient (see Section 2.6) it is worth mentioning a couple of possible variations. First of all, training the HPYLM on the Viterbi path can be seen as an approximation of the VB training. This approximation could be refined by sampling paths

instead of using the most likely one. Sampling several paths for an utterance would account for the uncertainty of the sequence unit. It was found experimentally that doing so considerably slows down the training and yields the same results as the method proposed above. Another important point is that we retrained from scratch the full HPYLM each time we update the HMMs' parameters. Indeed, the CRF assumes a fixed training data whereas in our case the sequences of units possibly change each time we update the acoustic model. This limitation could be tackled by removing all the customers of one utterance and then re-sampling a new sitting arrangement for this utterance. This approximation of the CRF is slightly inaccurate for very small data set but works well for any reasonable size data set. The possible speed up of this approximation is however counterbalanced by some memory overhead as we have to store the utterance corresponding to each customer in the CRF. No performance difference between the two approaches was found experimentally.

2.5 Evaluation

The evaluation of the discovered acoustic unit is not as straightforward as it may seem since the usefulness of the discovered units is highly task dependent. In this work, we use the mutual information between the human expert labeling and the discovered units. The mutual information between two random variables X and Y is defined as

$$I(X;Y) = H(X) - H(X|Y) \tag{2.1}$$

where $H(X)$ is the entropy of X and $H(X|Y)$ is the entropy of X given Y. Note that it is a symmetric measure. Informally, this metric can be understood as a "correlation" measure between the the discovered untis and the true phones. The mutual information gives a result in bits, however, since the maximum amount of bits to learn depends on the data and the task, we divide by the entropy of the true labels:

$$NMI = \frac{I(X;Y)}{H(X)} \tag{2.2}$$

where NMI stands for Normalized Mutual Information. This quantity is also known as the *uncertainty coefficient*. Note that the NMI version is not symmetric anymore and ranges from 0 to 1. Practically, we generate a sequence of units for each utterance of some test data using the Viterbi algorithm and then, we map each unit to its closest label in time. Using this one-to-one mapping the computation of the NMI is straightforward.

2.6 Results

The experiments were conducted on the TIMIT database [14]. We used two different sets of features: the mean normalized MFCC + Δ + $\Delta\Delta$ generated by HTK [15] and the Multilingual BottleNeck (MBN) features [16] trained on the Czech, German, Portuguese, Russian, Spanish, Turkish and Vietnamese data of the Global Phone database. As shown in Table 2.1, the bigram phone-loop model improves the NMI for both set of features. The improvement is relatively smaller with the MBN features. This is to be expected as the MBN features are trained and computed using some temporal context which reduces the influence of the bigram LM. Note that the results of the DP phone-loop model are slightly worse than the ones reported in [7] as we have used a separate test set rather than evaluating the NMI on the training data.

In standard ASR system, it is common practice to scale down the acoustic scores to alleviate the influence of the wrong assumptions of the HMM. Scaling down the acoustic score (in our case, this corresponds to multiplying Equation 5 in [7] by some scaling factor) reduces the dynamic range of the log-likelihood of the emissions' density and thus strengthens the influence of the state transitions and

model	features	NMI
DP phone loop	MFCC	33.94
Bigram phone loop	MFCC	**34.82**
DP phone loop	GP BN	42.06
Bigram phone loop	GP BN	**42.63**

Table 2.1: Normalized Mutual Information of the DP phone-loop and the bigram phone-loop for MFCC and MBN features

the language model. We found out experimentally that scaling the acoustic scores during the bigram phone-loop model training can significantly improve the final NMI. Figure 2.3 shows the absolute NMI improvement over the simple DP phone-loop model for various acoustic scale. The optimal scaling differs for the MFCC and the MBN features as the dynamic range of both feature set are rather different. Final results including the optimal acoustic scale for MFCC and MBN features are shown in

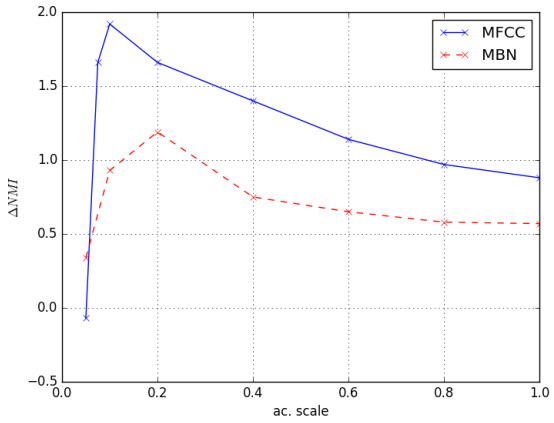


Figure 2.3: Absolute improvement of the NMI when scaling down the acoustic scores.

Table 2.2. For comparison, we computed the NMI from the output of a phone recogniser trained with

model	features	ac. scale	NMI
DP phone loop	MFCC	-	33.94
Bigram phone loop	MFCC	1.0	34.82
Bigram phone loop	MFCC	0.1	**35.86**
DP phone loop	GP BN	-	42.06
Bigram phone loop	GP BN	1.0	42.63
Bigram phone loop	GP BN	0.2	**43.25**
English phone rec.	-	-	42.21

Table 2.2: NMI of the DP phone-loop and the bigram phone-loop for MFCC and MBN features with optimal scaling

Kaldi [17] using the standard TIMIT recipe. Interestingly, the NMI of this baseline is similar to the MBN DP phone-loop and the bigram MBN phone-loop is about one percent better (see Table 2.2). Even though care has to be taken as the NMI is not a perfect metric, it is a promising result which let us hope that the research field of AUD will soon be mature enough to be applied to low-resource

languages that are so far out of reach of speech technologies.

2.7 Conclusion

We proposed to improve the AUD phone-loop model by incorporating a bigram language model. First, we train a DP phone-loop model to infer the number of units and then, the bigram phone-loop is initialized from the the DP phone-loop and trained until convergence of its parameters. Results show an improvement about 1-2 % of NMI for both MFCC and MBN features. When combined with MBN features, the AUD system has similar or higher NMI compared to a standard phone recogniser.

Chapter 3

Alternative inference methods for AUD model training

Yibo Yang, Lucas Ondel, and Lukáš Burget

The problem of automatic acoustic unit discovery (AUD) can be viewed as that of density estimation, i.e., learning or inferring a posterior distribution over the latent acoustic unit identities of speech observations. As exact inference is often intractable, two dominant approaches, Markov Chain Monte Carlo (MCMC) and Variational Bayesian methods (VB) have emerged to perform approximate inference in complex Bayesian models. Both approaches have their strengths and weaknesses, and an area of research has been developing methods that combine their strengths. This work investigates two such alternative inference methods applied to the finite Bayesian Gaussian mixture model, particularly the collapsed variational inference and collapsed Gibbs sampling algorithms, in the context of frame-level AUD. We show that compared to their un-collapsed versions, both algorithms experimentally lead to marginal improvements in acoustic clustering performance.

3.1 Introduction

Gibbs sampling (GS) is an MCMC algorithm that produces samples from the target posterior distribution by iteratively sampling from marginal distributions of its parameters; the samples from the true posterior is then used for inference tasks. As mentioned in 1.1.3, GS generally faces the issue of scalability, and requires monitoring of the convergence of the Markov chain. By contrast, VB is a deterministic algorithm that minimizes the divergence between the true posterior distribution and its approximation, wherein additional independence assumptions are made for tractability; the approximated true posterior is then used for inference. As seen in [7], VB training of a refined AUD model based on [6] led to much higher inference efficiency, as well as quality of inference (even though theoretically GS is guaranteed to produce samples from the true posterior, provided it reaches convergence).

Collapsed Gibbs sampling (collapsed GS) was introduced to address the inefficiency of GS by integrating out "nuisance" model parameters and only sampling from the much lower-dimensional space of latent variables. Collapsed variational Bayesian inference (collapsed VB) performs VB in the similarly lower dimensional space of a collapsed model, and aims to produce a more accurate approximation to the true posterior by making weaker independence assumptions. Depending on the exact parameters collapsed out, collapsed VB can retain the same computational advantage of standard VB, namely its amenability to parallel implementations, which can lead to higher quality, yet still scalable acoustic unit discovery.

To simplify the analysis, we use an approximation to the non-parametric Bayesian phone-loop

model introduced in Chapter 2; in particular we remove the HMM component, so that acoustic unit discovery is performed at frame-level (or, equivalently we restrict the HMM in the phone-loop model to having only one state), and we approximate the Dirichlet Process prior over the acoustic unit weights by a finite Dirichlet distribution.

In this chapter, we review the collapsed Gaussian mixture model and collapsed GS, fully derive the collapsed VB algorithm for GMM as proposed in [18], and perform experimental evaluations. In the derivation we adopt the notations in [18] and [19], and below is a list of frequently used notations and their definitions:

N	number of data points
D	dimensionality of data
$\mathbf{X} = \{\mathbf{x}_n \vert n \in [1, N]\}$	set of all data
$\mathbf{x}_n \in \mathbb{R}^D$	the nth data point
K	number of mixture components
$\mathbf{z} = \{z_n \vert n \in [1, N]\}$	set of latent variables responsible for data
$z_n \in [1, K]$	the nth discrete latent variable representing the id of the component responsible for $\mathbf{x}_n$
$\mathbf{z}^{\neg n} = \{z_m \vert m \in [1, N], m \neq n\}$	set of latent variables excluding the nth one
$\boldsymbol{\pi} = \{\pi_k \vert k \in [1, K]\}$	set of mixture component weights
$\boldsymbol{\eta} = \{\eta_k \vert k \in [1, K]\}$	set of mixture components' parameters
$\boldsymbol{\theta} = \{\boldsymbol{\eta}, \boldsymbol{\pi}\}$	set of all mixture model parameters
$\eta_k = \{\boldsymbol{\mu}_k, \boldsymbol{\Lambda}_k\}$	parameters of the kth component, i.e. mean vector and precision matrix of the kth Gaussian
$\boldsymbol{\mu} = \{\boldsymbol{\mu}_k \vert k \in [1, K]\}$	set of all means of Gaussian components
$\boldsymbol{\Lambda} = \{\boldsymbol{\Lambda}_k \vert k \in [1, K]\}$	set of all precisions of Gaussian components
$\boldsymbol{\alpha} = (\alpha_1, \alpha_2, \ldots, \alpha_K)$	parameters of Dirichlet prior on mixing weights
$\boldsymbol{\beta} = (\mathbf{m}_0, \beta_0, \nu_0, \mathbf{W}_0)$	parameters of Gaussian-Wishart prior on the mean and precision of each Gaussian component
$\mathbf{m}_0$	prior mean of $\boldsymbol{\mu}$
β_0	proportional to our belief in $\mathbf{m}_0$
ν_0	degree of freedom of Wishart prior on $\boldsymbol{\Lambda}$
$\mathbf{W}_0$	scale matrix of Wishart prior on $\boldsymbol{\Lambda}$

3.2 The Model

3.2.1 Finite Bayesian GMM

We model our data using a finite mixture of Gaussian distributions, with conjugate priors placed on mixture weights and Gaussian component parameters.

Each observed data point $\mathbf{x}_n$ is generated by the kth component with categorical probability $\pi_k = P(z_n = k|\boldsymbol{\pi})$ and multivariate Gaussian likelihood $p(\mathbf{x}_n|\eta_k) = \mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_k, \boldsymbol{\Lambda}_k)$, generally denoted $P(\mathbf{x}_n|\eta_{z_n}) = \mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_{z_n}, \boldsymbol{\Lambda}_{z_n})$ when the identity of the component z_n is not given.

We place a Dirichlet distribution prior on the mixing weights:

$$p(\boldsymbol{\pi}|\boldsymbol{\alpha}) = \mathrm{Dir}(\boldsymbol{\pi}; \alpha_1, \ldots, \alpha_K) = \frac{1}{C(\boldsymbol{\alpha})} \prod_{k=1}^{K} \pi_k^{\alpha_k - 1} \tag{3.1}$$

where $C(\boldsymbol{\alpha})$ is the normalizing constant: $C(\boldsymbol{\alpha}) = \frac{\prod_{k=1}^{K} \Gamma(\alpha_k)}{\Gamma(\hat{\alpha})}$ with $\hat{\alpha} = \sum_{k=1}^{K} \alpha_k$. The Dirichlet prior

is conjugate to the categorical (or equivalently, multinomial) likelihood of latent variables:

$$p(\mathbf{z}|\boldsymbol{\pi}) = \prod_{n=1}^{N} p(z_n|\boldsymbol{\pi}) = \prod_{k=1}^{K} \pi_k^{N_k} \tag{3.2}$$

where $N_k = \sum_n \mathbb{I}(z_n = k)$ is the number of data points belonging to component k. Often we use a symmetric Dirichlet prior[1] to indicate ignorance about mixing weights [19, (10.39)]:

$$p(\boldsymbol{\pi}|\boldsymbol{\alpha}) = \mathrm{Dir}(\boldsymbol{\pi}; \alpha_0, \ldots, \alpha_0) = \frac{1}{C(\boldsymbol{\alpha})} \prod_{k=1}^{K} \pi_k^{\alpha_0 - 1} \tag{3.3}$$

However to conserve space, we restrict our attention to the general case of (3.1), knowing that results for the symmetric case can always be obtained by replacing α_k with α_0 and $\hat{\alpha}$ with $K\alpha_0$. We also impose an independent Gaussian-Wishart prior on the mean and precision of Gaussian components [19, (10.40)]:

$$p(\boldsymbol{\eta}|\boldsymbol{\beta}) = \prod_{k=1}^{K} p(\eta_k|\boldsymbol{\beta}) = \prod_{k=1}^{K} \mathcal{N}(\boldsymbol{\mu}_k|\mathbf{m}_0, (\beta_0\boldsymbol{\Lambda})^{-1})\mathcal{W}(\boldsymbol{\Lambda}_k|\mathbf{W}_0, \nu_0) \tag{3.4}$$

which is fully conjugate to the data likelihood:

$$p(\mathbf{X}|\mathbf{z}, \boldsymbol{\eta}) = \prod_{n=1}^{N} p(\mathbf{x}_n|\eta_{z_n}) = \prod_{n=1}^{N} \mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_{z_n}, \boldsymbol{\Lambda}_{z_n}) \tag{3.5}$$

Thus the joint probability of our model is [2] [18, (7)]

$$p(\mathbf{X}, \mathbf{z}, \boldsymbol{\eta}, \boldsymbol{\pi}) = p(\mathbf{X}|\mathbf{z}, \boldsymbol{\eta})p(\mathbf{z}|\boldsymbol{\pi})p(\boldsymbol{\eta})p(\boldsymbol{\pi}) = \left[\prod_{n=1}^{N} p(\mathbf{x}_n|\eta_{z_n})p(z_n|\boldsymbol{\pi})\right]\left[\prod_{k=1}^{K} p(\eta_k)\right]\mathrm{Dir}(\boldsymbol{\pi}) \tag{3.6}$$

3.2.2 Collapsed Model

We integrate out the mixing weights from the above model and obtain the marginal probability of latent variable assignments [20, (24.24)]:

$$p(\mathbf{z}|\boldsymbol{\alpha}) = \int_{\boldsymbol{\pi}} p(\mathbf{z}|\boldsymbol{\pi})p(\boldsymbol{\pi}|\boldsymbol{\alpha})\,\mathrm{d}\boldsymbol{\pi} \tag{3.7}$$

$$= \frac{\Gamma(\hat{\alpha})}{\Gamma(N + \hat{\alpha})} \prod_{k=1}^{K} \frac{\Gamma(N_k + \alpha_k)}{\Gamma(\alpha_k)} \tag{3.8}$$

The joint probability of the collapsed model is therefore

$$p(\mathbf{X}, \mathbf{z}, \boldsymbol{\eta}) = p(\mathbf{X}|\mathbf{z}, \boldsymbol{\eta})p(\mathbf{z})p(\boldsymbol{\eta}) \tag{3.9}$$

$$= \left[\prod_{n=1}^{N} p(\mathbf{x}_n|\eta_{z_n})\right] p(\mathbf{z}) \left[\prod_{k=1}^{K} p(\eta_k)\right] \tag{3.10}$$

with $p(\mathbf{z})$ given above in (3.7).

[1] [18] observe that the symmetric Dirichlet distribution can well approximate the Dirichlet Process prior for large enough number of mixture components.

[2] We omit the hyperparameters $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ for brevity.

To prepare for later discussions, we calculate the probability of component assignment z_n conditioned all the other component assignments, i.e., the values of $\mathbf{z}_{\neg n}$, following [20, (24.25), (24.26)]:

$$p(z_n = k|\mathbf{z}_{\neg n}, \boldsymbol{\alpha}) = \frac{p(\mathbf{z}|\boldsymbol{\alpha})}{p(\mathbf{z}_{\neg n}|\boldsymbol{\alpha})} = \frac{N_k^{\neg n} + \alpha_k}{N^{\neg n} + \hat{\alpha}} \tag{3.11}$$

where $N^{\neg n}$ is the total number of data points excluding the nth, and trivially $N^{\neg n} = N - 1$.

We also calculate the posterior predictive distribution of the component id z_{new} associated with a new data point $\mathbf{x}_{new}$, given all the observed data and assignments. Following [20, (24.21)]:

$$\begin{aligned} p(z_{new} = k|\mathbf{z}, \mathbf{X}, \boldsymbol{\alpha}, \boldsymbol{\beta}) &\propto p(z_{new} = k|\mathbf{z}, \boldsymbol{\alpha}, \cancel{\boldsymbol{\beta}}) p(\mathbf{X}|z_{new} = k, \mathbf{z}, \cancel{\boldsymbol{\alpha}}, \boldsymbol{\beta}) && (3.12)\\ &= p(z_{new} = k|\mathbf{z}, \boldsymbol{\alpha}) p(\mathbf{x}_{new}|\mathbf{X}, z_{new} = k, \mathbf{z}, \boldsymbol{\beta}) && (3.13)\\ &\quad p(\mathbf{X}|\cancel{z_{new} = k}, \mathbf{z}, \boldsymbol{\beta})\\ &\propto p(z_{new} = k|\mathbf{z}, \boldsymbol{\alpha}) p(\mathbf{x}_{new}|\mathbf{X}, z_{new} = k, \mathbf{z}, \boldsymbol{\beta}) && (3.14) \end{aligned}$$

The first term comes from slightly modifying (3.11):

$$p(z_{new} = k|\mathbf{z}, \boldsymbol{\alpha}) = \frac{N_k + \alpha_k}{N + \hat{\alpha}} \tag{3.15}$$

To find the second term of (3.14), we use the fact that $p(\mathbf{x}_{new}|\mathbf{X}, z_{new} = k, \mathbf{z}, \boldsymbol{\beta}) = p(\mathbf{x}_{new}|\mathbf{X}_k, \boldsymbol{\beta})$, where $\mathbf{X}_k$ is all the data associated with component k, and the right hand side is the posterior predictive distribution of the kth Gaussian:

$$p(\mathbf{x}_{new}|\mathbf{X}_k, \boldsymbol{\beta}) = \int_{\eta_k} p(\mathbf{x}_{new}|\eta_k) p(\eta_k|\boldsymbol{\beta}) \, \mathrm{d}\eta_k \tag{3.16}$$

which can be shown to be a multivariate Student-T distribution:

$$p(\mathbf{x}_{new}|\mathbf{X}_k, \boldsymbol{\beta}) = \mathcal{T}(\mathbf{x}_{new}|\mathbf{m}_k, \frac{\beta_k(\nu_k - D + 1)}{\beta_k + 1}\mathbf{W}_k^{-1}, \nu_k - D + 1) \tag{3.17}$$

where $\mathbf{m}_k$, $\beta_k(\nu_k - D + 1)\mathbf{W}_k^{-1}/(\beta_k + 1)$, and $\nu_k - D + 1$ are respectively the mean, scale matrix, and degrees of freedom of the Student-T distribution; the updated parameters β_k, $\mathbf{m}_k$, $\mathbf{W}_k$, and ν_k are defined later in (3.47)-(3.50) in the context of variational inference.

3.3 Collapsed Gibbs Sampling

As much literature exists on collapsed Gibbs sampling for GMM, we refer the reader to [20] for detailed discussions. In fact, the full conditional distribution used in collapsed GS is identical to (3.12), with z_n and $\mathbf{z}_{\neg n}$ replacing z_{new} and $\mathbf{z}$.

3.4 Collapsed Variational Inference

3.4.1 Standard Factorization

The variational Bayesian inference algorithm lower bounds the log marginal likelihood of data with the negative variational free energy:

$$\begin{aligned} \mathcal{L}(q) &= \int\int q(\mathbf{z}, \boldsymbol{\theta}) \log \frac{p(\mathbf{X}, \mathbf{z}, \boldsymbol{\theta})}{q(\mathbf{z}, \boldsymbol{\theta})} \, \mathrm{d}\mathbf{z} \, \mathrm{d}\boldsymbol{\theta} && (3.18)\\ &= \log p(\mathbf{X}) + \int\int q(\mathbf{z}, \boldsymbol{\theta}) \log \frac{p(\mathbf{z}, \boldsymbol{\theta}|\mathbf{X})}{q(\mathbf{z}, \boldsymbol{\theta})} \, \mathrm{d}\mathbf{z} \, \mathrm{d}\boldsymbol{\theta} && (3.19)\\ &= \log p(\mathbf{X}) - KL(q(\mathbf{z}, \boldsymbol{\theta})||p(\mathbf{z}, \boldsymbol{\theta}|\mathbf{X})) && (3.20)\\ &\leq \log p(\mathbf{X}) && (3.21) \end{aligned}$$

where $\mathbf{z}$ is again the set of latent variables, $\boldsymbol{\theta}$ is the set of model parameters (which together with $\mathbf{z}$ are considered stochastic variables of the model in VB), $q(\mathbf{z}, \boldsymbol{\theta})$ is the variational approximation to the desired posterior distribution $p(\mathbf{z}, \boldsymbol{\theta}|\mathbf{X})$, and $KL(q(\mathbf{z}, \boldsymbol{\theta})||p(\mathbf{z}, \boldsymbol{\theta}|\mathbf{X}))$ is the Kullback-Leibler divergence from $p(\mathbf{z}, \boldsymbol{\theta}|\mathbf{X})$ to $q(\mathbf{z}, \boldsymbol{\theta})$, which is a non-negative measure of their dissimilarity.

In the standard mean-field setting, we assume independence between latent variables $\mathbf{z}$ and model parameters $\boldsymbol{\theta}$, giving rise to the following factorization

$$q(\mathbf{z}, \boldsymbol{\theta}) = q(\mathbf{z})q(\boldsymbol{\theta}) \tag{3.22}$$

Approximate inference is then achieved by maximizing the variational lower bound $\mathcal{L}$, which is equivalent to minimizing $KL(q(\mathbf{z}, \boldsymbol{\theta})||p(\mathbf{z}, \boldsymbol{\theta}|\mathbf{X}))$, alternately with respect to $q(\mathbf{z})$ and $q(\boldsymbol{\theta})$.

In the GMM of Section 3.2.1, $\boldsymbol{\theta} = \{\boldsymbol{\eta}, \boldsymbol{\pi}\}$, and factorization (3.22) becomes [18, (15)]

$$q(\mathbf{z}, \boldsymbol{\eta}, \boldsymbol{\pi}) = q(\mathbf{z})q(\boldsymbol{\eta}, \boldsymbol{\pi}) \tag{3.23}$$

$$= \left[\prod_{n=1}^{N} q(z_n)\right] \left[\prod_{k=1}^{K} q(\eta_k)\right] q(\boldsymbol{\pi}) \tag{3.24}$$

where the additional decompositions follow from the conditional indepences in the graphical model.

3.4.2 Collapsed Factorization

As noted in [18], the standard VB factorization (3.22) tends to be a bad assumption as it ignores the often strong dependence of latent variables $\mathbf{z}$ on the model parameters $\boldsymbol{\theta}$; the lower bound on log marginal likelihood can therefore be very loose and lead to inaccurate approximation.

In the original Gaussian mixture model, the latent variable assignments intimately depend on the mixing weights $\boldsymbol{\pi}$, which is ignored by factorization (3.23). We can finesse this problem by integrating out $\boldsymbol{\pi}$[3] and use the collapsed model introduced in Section 3.2.2. In the collapsed model, the set of model parameters $\boldsymbol{\theta}$ reduces to the set of mixture component parameters $\boldsymbol{\eta}$, and the dependence of $\mathbf{z}$ on $\boldsymbol{\pi}$ is preserved in the marginal distribution $p(\mathbf{z}|\boldsymbol{\alpha})$ given in (3.7), naturally resulting in the factorization $p(\mathbf{z}, \boldsymbol{\eta}) = p(\mathbf{z})p(\boldsymbol{\eta})$. Now applying (3.22) and assuming independence between $q(z_n)$ gives

$$q(\mathbf{z}, \boldsymbol{\eta}) = q(\mathbf{z})q(\boldsymbol{\eta}) \tag{3.25}$$

$$= \left[\prod_{n=1}^{N} q(z_n)\right] \left[\prod_{k=1}^{K} q(\eta_k)\right] \tag{3.26}$$

Note that collapsing introduces new dependency among the latent variables (which are previously conditionally independent), but since this dependency is spread out over a large number of latent variables, the factorization of $q(\mathbf{z})$ in (3.26) is a reasonable assumption.

3.4.3 Update Equations

As shown in [19, (10.9)], the log of the optimal form of each variational factor $q^*(\cdot)$ (which maximizes the lower bound $\mathcal{L}$) is obtained by taking the expectation of the log probability of the model joint distribution with respect to all the other variational variables. We apply this procedure to the collapsed GMM, keeping in mind that the optimal forms of $q(\mathbf{z})$ and $q(\boldsymbol{\eta})$ would be the same as the likelihood functions (i.e. categorical and Gaussian-Wishart) due to our use of conjugate priors.

[3] Ideally we would like to integrate out all the model parameters $\boldsymbol{\theta} = \{\boldsymbol{\eta}, \boldsymbol{\pi}\}$, so the fully collapsed VB algorithm would have a single update equation for the latent variable distribution $q^*(z_n)$; however the sequential nature of this algorithm (similar to that in collapsed Gibbs sampling) is an obstacle for parallel implementation and can make fully collapsed VB too computationally expensive.

Updating $q^*(z_n)$

The optimal form of $q(z_n)$, denoted $q^*(z_n)$, is given by

$$\log q^*(z_n) = \mathbb{E}_{\mathbf{z}^{\neg n},\boldsymbol{\eta}}\left[\log p(\mathbf{X},\mathbf{z},\boldsymbol{\eta})\right] + const. \tag{3.27}$$

$$= \mathbb{E}_{\mathbf{z}^{\neg n},\boldsymbol{\eta}}\left[\log p(z_n|\mathbf{z}^{\neg n}) + \log p(\mathbf{z}^{\neg n}) + \log p(\boldsymbol{\eta}) + \log p(\mathbf{X}|\mathbf{z},\boldsymbol{\eta})\right] + const. \tag{3.28}$$

where $\mathbf{z}^{\neg n}$ is the set of latent variables excluding the nth. Substituting (3.5) into $\log p(\mathbf{X}|\mathbf{z},\boldsymbol{\eta})$ and dropping terms that do not depend on z_n yields

$$\log q^*(z_n) = \mathbb{E}_{\mathbf{z}^{\neg n}}\left[\log p(z_n|\mathbf{z}^{\neg n})\right] + \mathbb{E}_{\eta_{z_n}}\left[\log p(\mathbf{x}_n|\eta_{z_n})\right] + const. \tag{3.29}$$

$$= \sum_{\mathbf{z}^{\neg n}} \prod_{m\neq n} q(z_m) \log p(z_n|\mathbf{z}^{\neg n}) + \int_{\eta_{z_n}} q(\eta_{z_n}) \log p(\mathbf{x}_n|\eta_{z_n})\, \mathrm{d}\eta_{z_n} + const. \tag{3.30}$$

Now we consider the two terms in (3.29) separately. In the first term, we recognize $p(z_n|\mathbf{z}^{\neg n})$ as the conditional distribution of latent variable z_n in the collapsed model; substituting in (3.11) for a particular value of $z_n = k$ gives

$$\mathbb{E}_{\mathbf{z}^{\neg n}}\left[\log p(z_n = k|\mathbf{z}^{\neg n})\right] = \mathbb{E}_{\mathbf{z}^{\neg n}}\left[\log\left(N_k^{\neg n} + \alpha_k\right)\right] + const. \tag{3.31}$$

For efficiency, we approximate the above expectation by the following second order Taylor expansion[4],

$$\mathbb{E}\left[f(m)\right] \approx f(\mathbb{E}\left[(m)\right]) + \frac{1}{2} f''(\mathbb{E}\left[m\right])\mathbb{V}\left[m\right] \tag{3.32}$$

(3.31) then becomes

$$\mathbb{E}_{\mathbf{z}^{\neg n}}\left[\log p(z_n = k|\mathbf{z}^{\neg n})\right] \approx \log\left(\mathbb{E}_{\mathbf{z}^{\neg n}}\left[N_k^{\neg n}\right] + \alpha_k\right) - \frac{\mathbb{V}_{\mathbf{z}^{\neg n}}\left[N_k^{\neg n}\right]}{2\left(\mathbb{E}_{\mathbf{z}^{\neg n}}\left[N_k^{\neg n}\right] + \alpha_k\right)^2} + const. \tag{3.33}$$

which can be efficiently computed by noting that the random variable $N_k^{\neg n}$ is the sum of (assumedly independent) Bernoulli random variables $N_k^{\neg n} = \sum_{m\neq n} \mathbb{I}(z_m = k)$, with mean and variance

$$\mathbb{E}_{\mathbf{z}^{\neg n}}\left[N_k^{\neg n}\right] = \sum_{m\neq n} q^*(z_m = k) \tag{3.34}$$

$$\mathbb{V}_{\mathbf{z}^{\neg n}}\left[N_k^{\neg n}\right] = \sum_{m\neq n} q^*(z_m = k)(1 - q^*(z_m = k)) \tag{3.35}$$

The second term of (3.29) for a particular $z_n = k$ is

$$\mathbb{E}_{\eta_k}\left[\log p(\mathbf{x}_n|\eta_k)\right] = \mathbb{E}_{\boldsymbol{\mu_k},\boldsymbol{\Lambda}_k}\left[\log \mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu_k},\boldsymbol{\Lambda}_k)\right] \tag{3.36}$$

$$= \frac{1}{2}\mathbb{E}_{\boldsymbol{\Lambda}_k}\left[\log\left|\boldsymbol{\Lambda}_k\right|\right] - \frac{1}{2}\mathbb{E}_{\boldsymbol{\mu_k},\boldsymbol{\Lambda}_k}\left[(\mathbf{x}_n - \boldsymbol{\mu_k})^\top \boldsymbol{\Lambda}_k (\mathbf{x}_n - \boldsymbol{\mu_k})\right] + const. \tag{3.37}$$

in which the expectation terms evaluate to [19, (10.64), (10.65)]:

$$\mathbb{E}_{\boldsymbol{\mu_k},\boldsymbol{\Lambda}_k}\left[(\mathbf{x}_n - \boldsymbol{\mu_k})^\top \boldsymbol{\Lambda}_k (\mathbf{x}_n - \boldsymbol{\mu_k})\right] = D\beta_k^{-1} + \nu_k(\mathbf{x}_n - \mathbf{m}_k)^\top \boldsymbol{\Lambda}_k (\mathbf{x}_n - \mathbf{m}_k) \tag{3.38}$$

$$\mathbb{E}_{\boldsymbol{\Lambda}_k}\left[\log\left|\boldsymbol{\Lambda}_k\right|\right] = \sum_{i=1}^{D} \psi\left(\frac{\nu_k + 1 - i}{2}\right) + D\log 2 + \log\left|\mathbf{W}_k\right| \tag{3.39}$$

where the terms β_k, $\mathbf{m}_k$, $\mathbf{W}_k$, and ν_k result from the updated component parameter distribution $q^*(\eta_k)$ to be discussed below, and are defined in (3.47)-(3.50).

[4]To see this, first write down the second order Taylor polynomial for $f(m)$ near the constant $\mathbb{E}\left[m\right]$:

$$f(m) \approx f(\mathbb{E}\left[m\right]) + f'(\mathbb{E}\left[m\right])(m - \mathbb{E}\left[m\right]) + \frac{1}{2} f''(\mathbb{E}\left[m\right])(m - \mathbb{E}\left[m\right])^2$$

then take expectations of both sides with respect to m and apply the definition $\mathbb{V}\left[m\right] = \mathbb{E}\left[(m - \mathbb{E}\left[m\right])^2\right]$

Updating $q^*(\eta_k)$

Similarly, for the optimal form $q^*(\eta_k)$ we have

$$\begin{aligned}\log q^*(\eta_k) &= \mathbb{E}_{\boldsymbol{\eta}^{\neg k},\mathbf{z}}\left[\log p(\mathbf{X},\mathbf{z},\boldsymbol{\eta})\right] + const. && (3.40)\\ &= \mathbb{E}_{\boldsymbol{\eta}^{\neg k},\mathbf{z}}\left[\log p(\eta_k) + \log p(\boldsymbol{\eta}^{\neg k}) + \log p(\mathbf{z}) + \log p(\mathbf{X}|\mathbf{z},\boldsymbol{\eta})\right] + const. \\ &= \mathbb{E}_{\mathbf{z}}\left[\log p(\eta_k) + \log p(\boldsymbol{\eta}^{\neg k}) + \log p(\mathbf{z}) + \log p(\mathbf{X}|\mathbf{z},\boldsymbol{\eta})\right] + const. && (3.41)\\ &= \log p(\eta_k) + \mathbb{E}_{\mathbf{z}}\left[\log p(\mathbf{X}|\mathbf{z},\boldsymbol{\eta})\right] + const. && (3.42)\end{aligned}$$

Here $p(\eta_k)$ is the joint probability of the mean and precision of the kth Gaussian given as factors in (3.4), and $\boldsymbol{\eta}^{\neg k}$ is the set of component parameters excluding the kth. We dropped $\boldsymbol{\eta}^{\neg k}$ from the expectation as each η_k is independent of another; we also dropped terms not dependent on η_k in (3.42).

To help simplify $\mathbb{E}_{\mathbf{z}}\left[\log p(\mathbf{X}|\mathbf{z},\boldsymbol{\eta})\right]$, we temporarily switch to an alternative representation of latent variables used in [19]: for each observation $\mathbf{x}_n$, the latent variable $\boldsymbol{\gamma}_n$ takes the form of a K dimensional one-hot-vector, with elements $\gamma_{nk} \in [0,1]$ for $k = 1,\ldots,K$. Now (3.5) is equivalent to

$$p(\mathbf{X}|\mathbf{z},\boldsymbol{\eta}) = \prod_{n=1}^{N}\prod_{k=1}^{K} p(\mathbf{x}_n|\eta_k)^{\gamma_{nk}} \tag{3.43}$$

therefore

$$\mathbb{E}_{\mathbf{z}}\left[\log p(\mathbf{X}|\mathbf{z},\boldsymbol{\eta})\right] = \sum_{n=1}^{N}\sum_{k=1}^{K}\mathbb{E}_{\mathbf{z}}\left[\gamma_{nk}\right]\log p(\mathbf{x}_n|\eta_k) = \sum_{n=1}^{N}\sum_{k=1}^{K} q^*(z_n=k)\log p(\mathbf{x}_n|\eta_k) \tag{3.44}$$

where we make use of the fact that $\gamma_{nk} = \mathbb{I}(z_n = k)$ is a Bernoulli random variable, whose mean with respect to the variational distribution $q^*(\mathbf{z})$ is simply the probability $q^*(z_n = k)$. Substituting (3.44) into (3.42) and again only keeping terms dependent on η_k gives

$$\log q^*(\eta_k) = \log p(\eta_k) + \sum_{n=1}^{N} q^*(z_n=k)\log p(\mathbf{x}_n|\eta_k) + const. \tag{3.45}$$

From the above, $q^*(\eta_k)$ can be shown to be an updated Gaussian-Wishart distribution:

$$q^*(\eta_k) = \mathcal{N}(\boldsymbol{\mu}_k|\mathbf{m}_k,(\beta_k\boldsymbol{\Lambda}_k)^{-1})\mathcal{W}(\boldsymbol{\Lambda}_k|\mathbf{W}_k,\nu_k) \tag{3.46}$$

where we define[5]

$$\beta_k = \beta_0 + \sum_{n=1}^{N} q^*(z_n=k) \tag{3.47}$$

$$\mathbf{m}_k = \frac{1}{\beta_k}\left[\beta_0\mathbf{m}_0 + \sum_{n=1}^{N} q^*(z_n=k)\mathbf{x}_n\right] \tag{3.48}$$

$$\mathbf{W}_k^{-1} = \mathbf{W}_0^{-1} + \beta_0\mathbf{m}_0\mathbf{m}_0^\top + \left[\sum_{n=1}^{N} q^*(z_n=k)\mathbf{x}_n\mathbf{x}_n^\top\right] - \beta_k\mathbf{m}_k\mathbf{m}_k^\top \tag{3.49}$$

$$\nu_k = \nu_0 + \sum_{n=1}^{N} q^*(z_n=k) \tag{3.50}$$

[5] [19, (10.61), (10.62)] give equivalent definitions for (3.48) and (3.49).

3.4.4 Evidence Lower Bound

We apply the definition of the lower bound (3.18) to the collapsed model, then decompose and rearrange terms:

$$\mathcal{L}(q) = \int q(\mathbf{z}, \boldsymbol{\eta}) \log \frac{p(\mathbf{X}, \mathbf{z}, \boldsymbol{\eta})}{q(\mathbf{z}, \boldsymbol{\eta})} \tag{3.51}$$

$$= \mathbb{E}_q \left[\log p(\mathbf{X}|\mathbf{z}, \boldsymbol{\eta})\right] - KL\left(q(\boldsymbol{\eta})||p(\boldsymbol{\eta})\right) + \mathbb{E}_q \left[\log p(\mathbf{z})\right] - \mathbb{E}_q \left[\log q(\mathbf{z})\right] \tag{3.52}$$

The first two terms can be shown [6] to take the simplified form

$$\mathbb{E}_q \left[\log p(\mathbf{X}|\mathbf{z}, \boldsymbol{\eta})\right] - KL\left(q(\boldsymbol{\eta})||p(\boldsymbol{\eta})\right] = -\frac{DN}{2}\log 2\pi + \frac{DK}{2}\log\beta_0 - \frac{D}{2}\sum_{k=1}^{K}\log\beta_k$$

$$+ K\log B(\mathbf{W}_0, \nu_0) - \sum_{k=1}^{K}\log B(\mathbf{W}_k, \nu_k) \tag{3.53}$$

where $B(\mathbf{W}, \nu)$ is the normalizing constant of the Wishart distribution given in [19, (B.79)].

For the third term of (3.52), substituting in (3.8) gives

$$\mathbb{E}_q \left[\log p(\mathbf{z})\right] = \log\Gamma(\hat{\alpha}) - \log\Gamma(N + \hat{\alpha}) - \sum_{k=1}^{K}\log\Gamma(\alpha_k) + \sum_{k=1}^{K}\mathbb{E}_q \left[\log\Gamma(N_k + \alpha_k)\right] \tag{3.54}$$

where the expectation term can again be approximated by the second order Taylor polynomial (3.32):

$$\mathbb{E}_q \left[\log\Gamma(N_k + \alpha_k)\right] \approx \log\Gamma\left(\mathbb{E}_{\mathbf{z}}\left[N_k\right] + \alpha_k\right) + \frac{1}{2}\psi_1\left(\mathbb{E}_{\mathbf{z}}\left[N_k\right] + \alpha_k\right)\mathbb{V}_{\mathbf{z}}\left[N_k\right] \tag{3.55}$$

where $\psi_1(\cdot)$ is the trigamma function, and $\mathbb{E}_{\mathbf{z}}\left[N_k\right]$ and $\mathbb{V}_{\mathbf{z}}\left[N_k\right]$ can be calculated in the same manner as (3.34) and (3.35).

The last term of (3.52) is simply $\mathbb{E}_q \left[\log q(\mathbf{z})\right] = \sum_{n=1}^{N}\sum_{k=1}^{K} q^*(z_n = k)\log q^*(z_n = k)$.

3.4.5 A Note on Implementation

Collapsed VB can be implemented with only minor modifications to an existing implementation of standard VB, and both have the same computational complexity. In VB, we cycle between two stages analogous to the E and M steps of the EM algorithm, with one stage dependent on the results of the other. In collapsed VB, we no longer maintain the weights $\boldsymbol{\pi}$, and to calculate $q_t^*(z_n)$ in the variational E-step at time t, we not only require distributions over model parameters from the previous M-step $q_{t-1}^*(\eta_k)$, but also the same distribution of latent variables from the previous E-step $q_{t-1}^*(z_n)$.

3.5 Experimental Evaluation and Conclusion

We implemented GS, VB, and their collapsed versions in Python. We first used them to cluster synthetic data, with a typical data set of size 200 and consisting of 30-dimensional data generated by a 30-component Gaussian mixture. In all the algorithms, we set the number of model components K to 10 plus that of the data-generating GMM; we initialize the models by randomly assigning data to the components (with the same seed); we also use the same prior setting, with $\mathbf{m}_0$ and $\mathbf{W}_0$ set to the mean and covariance of the data, ν_0 set to the data dimension plus 2, and β_0 and all α_k set to 1. During training, we monitored the marginal joint probability of data and their assignment $p(\mathbf{X}, \mathbf{z})$ in (collapsed) GS, as well as the likelihood lower bound on the data in (collapsed) VB, with typical results shown in Figure 3.1 and Figure 3.2. We observed that collapsed GS often mixed faster, using

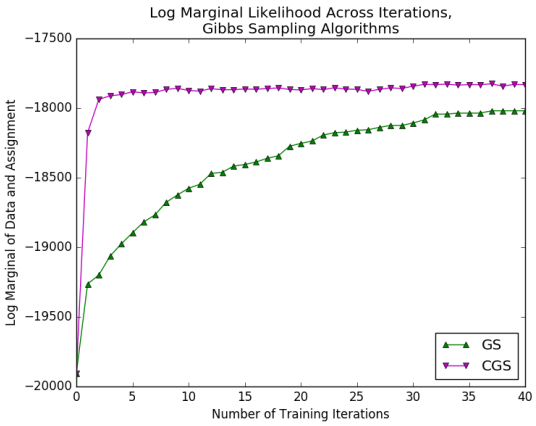


Figure 3.1: Collapsed vs. standard GS

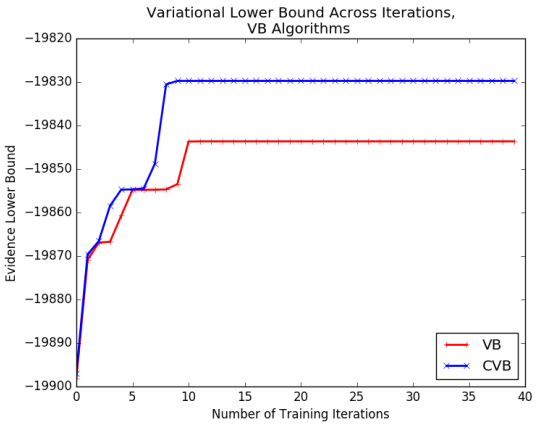


Figure 3.2: Collapsed vs. standard VB

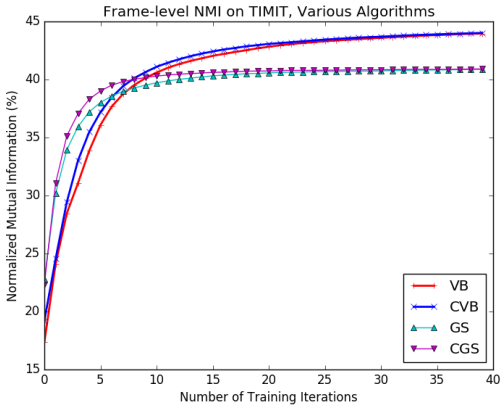


Figure 3.3: Frame-level NMI, TIMIT

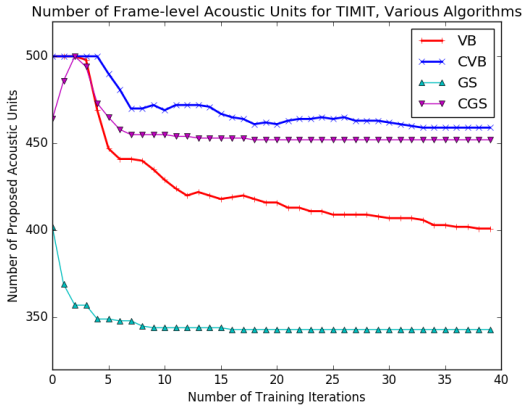


Figure 3.4: Number of acoustic clusters, TIMIT

many fewer iterations than standard GS; similarly, collapsed VB typically enjoyed faster convergence and tighter lower bound than standard VB.

We also used these algorithms to perform frame-level AUD on TIMIT. We trained the models on 39-dimensional MFCC features, labeled the test data by calculating (3.14)-(3.17), and evaluated Normalized Information using frame-level transcripts. We initialized the models by the same procedure used above for clustering synthetic data, except we set the number of Gaussian components to 500 [7]. As seen in Figure 3.3, the collapsed algorithms converged slightly faster than their un-collapsed versions, and resulted in 1-2 % higher NMI in early stages of training. Interestingly, as seen in Figure 3.4, the collapsed algorithms also had the tendency to produce more acoustic clusters. We also observed little to no difference between using Gaussian components with full or diagonal covariance matrices, as well as between using only the first or both terms in the approximation (3.32) for collapsed VB [8]. However, collapsed GS on average took 25 % more CPU time per iteration than standard GS.

In conclusion, we derived the collapsed variational inference algorithm for finite GMM, experimented with it along with collapsed Gibbs sampling for acoustic clustering, and found marginal improvements over their standard counterparts.

[6] `http://utdallas.edu/~yxy142230/notes/simplified_vb_lower_bound_gmm.pdf`

[7] We empirically found this upper bound on the effective value of K by counting the number of unique test data labels after several experiments with various K, as well as by performing the same clustering with Dirichlet Process GMM.

[8] Unsurprisingly, the former approximation of only using first-order terms resulted in slightly less CPU time than both standard VB (1 % on average) and than if both terms are used (4 % on average).

Chapter 4

Multilingual Acoustic Unit Discovery

Leda Sarı, Lucas Ondel, Lukáš Burget

Assuming that there is a common underlying generative model for the acoustic units in languages and some acoustic units are shared between languages we can make use of the high-resource languages which have sufficient amount of data to train more reliable systems. Therefore we can improve the acoustic unit discovery (AUD) performance in a resource-less language by transferring knowledge from other languages. This chapter presents methods that either use the posterior estimates of the parameters in AUD models trained on different languages as the prior in the target language AUD training procedure or utilize information contained in multiple languages while extracting features of the target data for AUD training.

4.1 Introduction

Given the fact that there is abundance of devices that have audio recording capabilities, it is easy to collect speech resources. However, training reliable systems with supervision requires having large amount of transcribed data. The task of transcription by human experts is an expensive task and in some cases such as for endangered languages, it might be impossible. Although each language can have distinctive phonetic units, some of them are shared across languages. Therefore, several multilingual approaches are proposed to make use of this common underlying structure in acoustic modeling [1, 129]. If we assume that there is a common generative model that covers multiple languages, then high-resource languages can be utilized in acoustic unit discovery (AUD) task.

The aim of this chapter is to use multilingual data to improve the acoustic units discovered by the variational Bayesian inference technique of [23] and the methods described in Chapter 2. The information can be transferred to the target data either by using it as the prior knowledge of the model or by obtaining a language-independent data representation of the target data that captures the common information in languages and then using this representation as the input to the AUD system.

This chapter is organized as follows: The methods to transfer information from one language to another is summarized in Section 4.2, then the experimental setup and results are presented, the chapter is concluded by summarizing the findings.

4.2 Methods

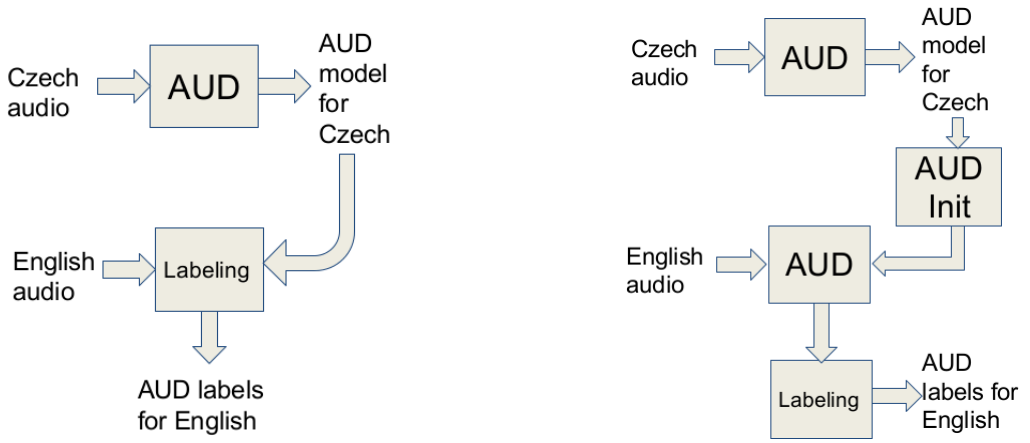
This section summarizes the ways of transferring information from multilingual datasets to the resource-less target language. Depending on the use of available transcribed data for resourceful

languages, three methods are used which can be grouped into two main categories:

1. Completely unsupervised approaches
    - Direct use of an AUD model from other languages
    - Transferring the posterior estimates of AUD parameters as the prior
2. Supervised approach via multilingual bottleneck features (BNF)

In the first approach, only audio from different languages are used to train AUD models. The most basic way of utilizing an AUD model on another language is directly using the model to decode the target data. In this method, a model from a mismatched language is used in Viterbi decoding phase described in Section 2.2. Here it is assumed that both languages share the common units or units in one language is sufficient to express units in the other. However, this assumption will not completely be true as there can exist phonetic units which does not exist in the other ones. Therefore, a better way is to transfer knowledge from this AUD model and train another AUD model on the target data based on that knowledge, possibly allowing the discovery of the additional units that only exist in the target language.

As described in [23], the training procedure of the AUD model consists of estimating the posterior distribution of the hidden variables such as the HMM states and GMM component associated with each frame and parameters such as transition probabilities. In the Variational Bayesian framework of [23], the hidden variables and parameters are assumed to be independent and conjugate priors are used so that the posterior estimates of the variables (or parameters) have closed form solution. These priors have hyper-parameters that have to supplied at the beginning of training which are chosen heuristically. However, if we have knowledge from another language, then more informative priors can be used to initialize the training. Therefore, the second way of making use of multilingual data in an unsupervised fashion is to get the posterior estimates of the parameters from one language and then use it as the prior for training another AUD model on the target data. Figure 4.1 summarizes these two unsupervised approaches when we transfer knowledge from Czech to English.



(a) Direct use of the AUD

(b) Posteriors to priors

Figure 4.1: Direct use of the AUD model and using posteriors from Czech to initialize the priors of the AUD training of English

The above mentioned approaches do not fully exploit the resources in other languages since they only use the audio components. In the second approach, there is supervision where available written resources are also utilized. In this approach, the transcriptions are used to train a multilingual bottleneck network [16], then the input features for AUD training on the target data are extracted from

the bottleneck layer of this network. These bottleneck features (BNF) give a language-independent representation of the audio and it makes use of the information hidden in multilingual dataset. Besides supervision, the second difference of this method from the unsupervised ones described above is that here the knowledge transfer is achieved at the feature level instead of the model level.

4.3 Experiments and Results

In the experiments, Wall Street Journal (WSJ) dataset [125, 126] is used as the target language. Seven languages (Czech, German, Portuguese, Russian, Spanish, Turkish and Vietnamese) from the GlobalPhone (GP) dataset [98] are chosen as our multilingual data. The AUD performance is measured using normalized mutual information (NMI) presented in Section 2.5. In addition, the number of units used to express the test data is also reported.

Table 4.1 summarizes the results for the completely unsupervised setups. The second and third columns show the performance when we train an AUD model from a language and directly use it to decode the target language, which is English in our case. The last two columns show the NMI and the number of units when we use the posterior estimates of the parameters for a language as our prior for the target. Except for the last line of the table, the information is transferred from a single language to the target language. In the experiment labeled as '7 languages' equal-sized subsets of each language are combined into a single dataset such that the total duration of this mixed dataset is approximately matches the duration of training data in the individual languages which is about 20 hours per language.

When we directly use a mismatched model to decode WSJ which does not transfer knowledge indeed, the performance is lower than that with a model trained on the same corpus as expected. However, this experiment serves as a baseline to show whether we can improve NMI if we use posteriors of AUD model parameters as our prior on WSJ. If we transfer knowledge from posterior to priors and train an AUD model on WSJ with these priors, higher NMI is achieved as compared to the direct use of models as shown in Table 4.1. However, except for Czech (28.17%) and German (28.78%), this type of transfer does not improve the performance as compared to using the matched condition where the AUD is performed only on the target data (28.12%). It is also observed that in Czech and German experiments, we start with certain units but as we train on the target data, the number of decoded units increase after using the informative priors.

	Direct use		Posterior to prior	
Language	NMI	# of units	NMI	# of units
English (WSJ)	28.12	81	-	-
Czech	26.47	73	28.17	74
German	26.51	73	28.78	77
Portuguese	26.28	81	27.77	81
Russian	25.55	79	27.32	79
Spanish	25.51	75	27.34	75
Turkish	26.00	73	27.70	73
Vietnamese	24.06	82	26.76	82
7 languages	26.26	78	27.37	78

Table 4.1: NMI (in %) and the number of discovered acoustic units for WSJ when AUD models from different languages are used in an unsupervised manner

Table 4.2 summarizes the NMI where we make use of the multilingual dataset while extracting our features for the target data which are in turn used as input to AUD training. If we use BNF for

WSJ to train an AUD model, we observe 28.8% relative improvement over using MFCCs. If we also incorporate the bigram language modeling approach described in Section 2.4, we get slightly lower NMI than the unigram language model but it still performs better than the case where we use MFCC features. Therefore, the main improvement comes from the input features.

Feature	LM	NMI	# units
MFCC	Unigram	28.12	81
BNF	Unigram	36.21	95
BNF	Bigram	36.15	95

Table 4.2: NMI (in %) and the number of discovered acoustic units for WSJ when BNF extracted from multilingual neural networks are used along with different LM strategies in AUD

4.4 Conclusions

In the experiments, it is observed that following a fully unsupervised approach which does not make use of the transcribed data of the multilingual dataset but makes use of the posterior estimates of AUD parameters in one language as a prior on the target language, does not improve NMI. On the other hand, utilizing the available transcribed data to train a multilingual bottleneck network and then using the BNF of the target data as our input features to the AUD training led to 8.1% absolute (28.8% relative) improvement in NMI. Therefore, finding a language-independent representation of the target data or exploiting supervised data on unrelated languages is a way to improve the AUD performance measured in terms of NMI.

Chapter 5

An Empirical Evaluation of Zero Resource Acoustic Unit Discovery

Chunxi Liu, Jinyi Yang, Ming Sun, Santosh Kesiraju, Alena Rott, Lucas Ondel, Pegah Ghahremani, Najim Dehak, Lukaš Burget and Sanjeev Khudanpur

Acoustic unit discovery (AUD) is a process of automatically identifying a categorical acoustic unit inventory from speech and producing corresponding acoustic unit tokenizations. AUD provides an important avenue for unsupervised acoustic model training in a zero resource setting where expert-provided linguistic knowledge and transcribed speech are unavailable. Therefore, to further facilitate zero-resource AUD process, in this chapter, we demonstrate acoustic feature representations can be significantly improved by (i) performing linear discriminant analysis (LDA) in an unsupervised self-trained fashion, and (ii) leveraging resources of other languages through building a multilingual bottleneck (BN) feature extractor to give effective cross-lingual generalization. Moreover, we perform comprehensive evaluations of AUD efficacy on multiple downstream speech applications, and their correlated performance suggests that AUD evaluations are feasible using different alternative language resources when only a subset of these evaluation resources can be available in typical zero resource applications.

5.1 Introduction

Standard supervised training of automatic speech recognition (ASR) systems typically replies on transcribed speech audio and pronunciation dictionaries. However, for a large majority of the world's languages, it is often difficult or even almost impossible to collect enough language resources to develop ASR systems with current standard ASR technology [21]. Therefore, developing speech technologies for a target language with zero expert-provided resources in that language becomes a significant challenge.

Recent zero resource efforts focused on phonetic discovery, or acoustic unit discovery (AUD), have made important progress in fully unsupervised acoustic model training and performing subword unit tokenization [22, 23]. In [22], a Dirichlet process hidden Markov model (DPHMM) framework is formulated to simultaneously perform three sub-tasks of segmentation, nonparametric clustering and sub-word modeling, and the spoken term detection task is used to evaluate the learned sub-word models. [23] also presents a nonparametric Bayesian framework to solve the same problem of unsupervised acoustic modeling with three major differences: (i) the Gibbs Sampling (GS) training algorithm is replaced with Variational Bayesian (VB) inference, which allows parallelized training amenable to large scale applications, (ii) a phone-loop model with a mixture of HMMs (each phone-

like acoustic unit is modeled by a HMM) is seen as a single HMM and thus does not require sub-word boundary variables, and (iii) normalized mutual information (NMI) between the hypothesized acoustic unit sequences and orthographic phoneme transcripts is used to evaluate the modeling efficacy.

As being unknown to the guidance of word transcripts in zero-resource scenarios, effective acoustic front-end processing becomes particularly critical to uncover the phonetic salience by the acoustics themselves. In supervised ASR system, linear discriminant analysis (LDA) [24] is often employed to exploit substantial contextual information, and the target class labels for LDA can be context-dependent triphone states given by the forced alignments of speech transcripts that are unavailable in zero resource setting. In this Chapter, we explore applying similar LDA strategy but with target labels acquired by the first-pass acoustic unit tokenizations, which is considered as a self-supervised fashion to solve the unknown label problem. Previous work in [25] also exploits such unsupervised LDA to support Dirichlet process Gaussian mixture model (DPGMM) based clustering although being limited to frame-level clustering without acoustic unit-level segmentation.

To date language-independent bottleneck (BN) features have been demonstrated as effective speech representations in improving ASR accuracies [26, 27]. In our study, we explore a state-of-the-art multilingual time delay neural network (TDNN) technique to generate robust cross-lingual acoustic features in zero-resource setting, and the hope is that as one moves to new languages, this data driven feature extraction approach will work as-is, without having to redesign feature extraction algorithms.

In this chapter, we employ the AUD framework in [23] and investigate the efficacy of incorporating LDA and multilingual BN TDNN techniques to AUD. Given the two distinct evaluations in [22, 23], we proceed by conducting not only an intrinsic measure of assessing the NMI between model hypothesis and true reference, but also an extrinsic measure of AUD's utility to downstream speech tasks.

Past studies in [28, 29] demonstrated the effectiveness of posterior features based on automatically derived acoustic structures by spoken term detection and phoneme discrimination tasks, while being limited to frame-level clustering and loss of phonetic temporal information. In contrast, [22] succeeded in computing posteriorgram representations over the learned sub-word units, capturing the phonetic context knowledge. In this chapter, we also exploit the feature representation of posteriorgrams across acoustic units learned from our AUD procedure, and test by a unified evaluation framework proposed in [30, 31] that quantifies how well speech representations enable discrimination between word example pairs of the same or different type, which is referred to as the same-different task and characterized by average precision (AP). [30] demonstrates almost perfect correlation between such AP and phone recognition accuracies of supervised acoustic models; therefore, we would like to investigate if such AP can also be a proxy for the unsupervised AUD accuracies, such that we can still evaluate AUD efficacy in the zero-resource condition that no orthographic phoneme transcripts for NMI measure are available but only word pairs. Since such word pairs can not only be obtained from manual transcripts, also from unsupervised spoken term discovery systems without relying on any language-specific resources [32, 33], at little cost compared with expensive phoneme transcripts.

Finally, previous work like [34, 35] presented the success of using unit- or word-level acoustic patterns discovered from fully unsupervised setting to provide competitive performance in spoken document topical classification and clustering, we also explicitly measure our AUD utility of learning document representations in this study.

5.2 Improving Feature Representation Learning for acoustic unit discovery

AUD is to discover repeated acoustic patterns in the raw acoustic stream and learn speaker independent acoustic models for each unique acoustic unit. We employ the same nonparametric Bayesian framework as [23]. A phone-loop model is developed as shown in Figure 5.1, and each unit is modeled as a Bayesian GMM-HMM. Under Dirichlet process framework, we consider the phone-loop as an infinite mixture

of GMM-HMMs, and the mixture weights are based on the stick-breaking construction of Dirichlet process. Following [36], the infinite number of units in the mixture is approximated by a truncation number T, giving zero weight to any unit greater than T.

Model parameters are fully Bayesian, with corresponding prior and posterior of conjugate distributions for each parameter. The Variational Bayesian (VB) inference (seen as an extension of the expectation-maximization algorithm that computes posterior distributions of both model parameters and latent variables) [37] is used to train the full Bayesian models. We initialize the hyperparameters for the prior distributions, and the posterior distributions are initialized the same as their prior distributions before the first training iteration starts. During each iteration, sufficient statistics are computed and accumulated to update the model posterior distributions. We can treat such mixture of GMM-HMMs as a single unified HMM with loop transitions, and thus the segmentation of the data can be performed using standard forward-backward algorithm in an unsupervised fashion. Parallelized training is conducted and convergence monitored by computing a lower bound on the data log-likelihood. After VB training, during evaluation we use Viterbi decoding algorithm to obtain acoustic unit tokenizations of the data, or forward-backward algorithm to produce posteriorgrams across the learned acoustic units.

5.2.1 LDA with Unsupervised Learning

We first parameterize the acoustic data into Mel-frequency cepstral coefficients (MFCCs) or BN features and apply Cepstral mean and variance normalization (CMVN), perform a first-pass AUD training over such raw acoustic features, obtain acoustic unit HMM state tokenizations of the data, i.e., 1-best HMM state-level decode for each acoustic frame, and use the resulting state-level labels as the class labels for LDA.

To apply LDA, additional context frames after CMVN are stacked to around the center frame. LDA is then performed on this higher-dimensional, context-rich representation. We apply the resulting LDA transformation to project the context-rich raw acoustic features back into a lower dimensional representation. These vectors after CMVN are subsequently used for a second-pass AUD training. Note that, for the second-pass VB training on the LDA-based features, rather than starting from scratch, we can first use the models learned from first-pass training to compute certain sufficient statistics that can be transferred regardless of different front-end features, and use them to update the model posteriors just for the first iteration; e.g., we can transfer the MFCC/BN-based statistics of accumulated posteriors of certain latent variables (acoustic unit, HMM state or GMM component), and use them to re-estimate the posterior's parameters in the first iteration of LDA-based training, by assuming certain acoustic structures discovered by the first-pass model being more accurate than those by our prior models.

5.2.2 Cross-lingual Generalization of Multilingual BN Network

In our multilingual BN training recipe, we use the TDNN architecture with parallel GPU training (using up to 8 GPUs) as described in [38] with two major extensions. First, hidden layers with ReLU nonlinearity are shared across languages (ReLu dimension 600, i.e., the output dimensions of the weight matrices), while separate language-specific final output layers with context-dependent triphone state targets are used for each different language. Second, an additional 42-dimensional bottleneck layer is added just before the final output layers, giving 6 hidden layers in total. Moreover, 3-fold training data augmentation with speed perturbations of 0.9, 1.0 and 1.1 are used. Each mini-batch of training data is randomly sampled based on the relative amounts of acoustic data in different languages, and any data of each language is used only once in one epoch. 40-dimensional MFCCs (without cepstral truncation [38]) augmented with 3-dimensional pitch and probability of voicing features are used as inputs to the network.

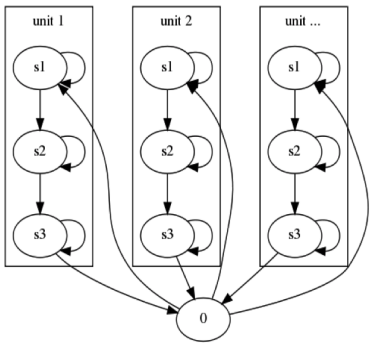


Figure 5.1: AUD phone-loop model with an infinite number of units and each unit modeled by a Bayesian GMM-HMM.

We developed our TDNN-based BN training and validation using multiple languages in both hybrid and tandem HMM-based ASR systems, while being unknown to the target language on which we perform AUD. We assume the word error rate reductions in our BN-based ASR tasks will translate into more effective cross-lingual generalization of our BN techniques on unseen target language, in turn, facilitating more accurate AUD.

5.3 Evaluating Acoustic Unit Discovery

5.3.1 NMI against Orthographic Phoneme Transcripts

After VB training of the Bayesian AUD models, to evaluate the quality of the automatically learned acoustic models, we first obtain acoustic unit tokenizations, i.e., 1-best HMM unit-level decode, of the development data on which AUD training is performed; alternatively, we can also use the learned models to obtain tokenizations of any evaluation data that the models do not see during training. Then we align the decoded acoustic unit sequence $\mathbf{Y}= Y_1, ..., Y_N$ with reference phoneme sequence $\mathbf{X}= X_1, ..., X_M$, and each $Y_j (1 \leq j \leq N)$ is aligned to a $X_i (1 \leq i \leq M)$, based on which the mutual information $I(\mathbf{X}; \mathbf{Y})$ is computed. We normalize it by the entropy $H(\mathbf{X})$ of $\mathbf{X}$, giving the normalized mutual information $NMI = I(\mathbf{X}; \mathbf{Y})/\mathbf{H}(\mathbf{X})$. $NMI = 0$ means $\mathbf{Y}$ carries no information about $\mathbf{X}$, and $NMI = 1$ means $\mathbf{Y}$ perfectly predicts $\mathbf{X}$.

5.3.2 Same-Different Evaluation

To evaluate AUD models, we can also apply them to a data set by using forward-backward algorithm to compute posterior distributions across the learned acoustic units over time. Thus, any word segments required by the same-different task can be given such HMM unit-level or state-level posteriorgram features. For each word pair, we compute a pairwise normalized dynamic time warping (DTW) distance with symmetric KL-Divergence as frame-level distance metric; since our acoustic features are posterior distributions, symmetric KL divergence are demonstrated superior to cosine distance for posterior features [30]. The pairwise normalized DTW distance is further used as a same/different classifier score; if the score is lower than some threshold τ, we declare this word pair corresponds to the same word type. As we sweep the threshold τ, we can obtain a standard precision-recall curve, under which the area is computed as the average precision (AP). In such means, we investigate if the better posterior estimates across automatically derived acoustic categories in AUD procedure can translate into the improved discriminability of separating same word type pairs from different word type pairs.

5.3.3 Spoken Document Classification and Clustering

Spoken document topical classification/identification (ID) is to classify a given document into one of the predefined set of topics or classes. Typically, documents are characterized based on a bag-of-words multinomial representation [39], or a more compact vector given by probabilistic topic models [40]. To evaluate the quality of the acoustic unit tokenizations of spoken documents, we employ the document representations as bags of acoustic units. For such classification task with topic labeled training data, we use stochastic gradient descent based linear SVM [41, 42] as our multi-class classifier training algorithm, with hinge loss and $\mathcal{L}^1$ norm regularization.

In the case that no topic labels are available, we can still perform unsupervised document clustering by the bags of acoustic units representation. We would like to investigate if reasonable clustering performance can be obtained without using manual or automatic transcript from supervised acoustic models but only unsupervised AUD. Following [35], we use the clustering algorithm of globally optimal repeated bisection [43].

5.4 Experiments

5.4.1 Experimental Setup

For our experiments we use the Switchboard Telephone Speech Corpus [44], a collection of two-sided telephone conversations with a single participant per side. Following the data set split strategy in [35], we use the same development and evaluation data set as [35]. There are 360 conversation sides of six different topics (recycling, capital punishment, drug testing, family finance, job benefits, car buying) in the development data set of 35.7 hours of audio. Each conversation side (seen as a single document) has one single topic, and each topic has equal number of 60 sides of conversations. Similarly, there are another different six topics (family life, news media, public education, exercise/fitness, pets, taxes) evenly across the 600 conversation sides of evaluation data set (61.6 hours of audio). Unsupervised VB training of acoustic unit models are performed on the development set (10 iterations); after the unsupervised learning, we apply the learned acoustic unit models to obtain the acoustic unit tokenizations of both development and evaluation sets.

For AUD model definitions, we use the truncation $T = 200$, which implies maximum 200 different acoustic units can be learned from the corpus. For each acoustic unit, we use a HMM of 3 emission states with a left-to-right topology and 2 Gaussians per state. Other hyperparameter values are the same as [23].

To compute NMI, we first use a supervised ASR system trained on Switchboard training corpus (about 300 hrs) to obtain forced aligned phoneme transcripts as our reference transcripts. During scoring, we define the distance between an output acoustic unit token and a reference phoneme token as the time frame difference between the center frames of two tokens; in doing so, each acoustic unit token is assigned to a closest reference phoneme token based on the distance metric defined. As shown in Table 5.1, the number of units in the tokenizations of a dataset is determined as the number of unique units that occur in any of the 1-best Viterbi decode of that dataset; thus, truncation $T = 200$ is the ceiling number, and it is possible that unit numbers differ between development and evaluation data since all 200 unit models are used during decoding process.

5.4.2 Feature Extraction Using LDA and Multilingual BN

For AUD experiments, we use manual segmentations provided by the Switchboard corpus to produce utterances with speech activity, and speech utterances are further parameterized either as 39-dimensional MFCCs with first and second order derivatives, or 42-dimensional BN features, with CMVN applied per conversation side.

Table 5.1: *AUD Performance evaluated by NMI, same-different task, document classification and clustering on Switchboard*

Dataset	Acoustic Features AUD is based on	# units	% NMI	Average Precision	Document Classification Accuracy	Document Clustering Purity	Document Clustering B-Cubed F1
Development Data	MFCC	145	21.59	0.247	0.3083 ± 0.0908	0.2268 ± 0.0015	0.1817 ± 0.0008
	MFCC w/ LDA	145	24.55	0.251	0.4361 ± 0.0692	0.2354 ± 0.0026	0.1855 ± 0.0006
	BN	184	28.20	0.343	0.7028 ± 0.0796	0.2446 ± 0.0018	0.1949 ± 0.0008
	BN w/ LDA	184	29.13	0.359	0.7167 ± 0.0733	0.2553 ± 0.0102	0.2023 ± 0.0047
Evaluation Data	MFCC	144	21.20	0.224	0.4633 ± 0.0702	0.2388 ± 0.0010	0.1899 ± 0.0001
	MFCC w/ LDA	144	24.07	0.219	0.4833 ± 0.0477	0.2426 ± 0.0031	0.1893 ± 0.0005
	BN	184	28.01	0.303	0.7167 ± 0.0350	0.2398 ± 0.0069	0.1983 ± 0.0032
	BN w/ LDA	184	28.84	0.329	0.7300 ± 0.0567	0.2373 ± 0.0037	0.2140 ± 0.0035

11-frame context windows of raw acoustic features (MFCCs or BN features) with CMVN are stacked to represent the center frame (equal left and right context frames as 5), and used as the LDA inputs. Using truncation parameter $T = 200$ and 3 HMM emission states yields 600 possible unique HMM state labels for the first-pass tokenization of development data. These state labels are used as LDA class labels. We accumulate LDA statistics and estimate the transformation matrix from development data, and apply the resulting LDA transformation to both development and evaluation data, reducing the spliced raw acoustic features into 40 dimensions for each frame. Then we proceed with second-pass AUD training based on the 40-dimensional LDA features. We reuse the sufficient statistics of certain latent variables (i.e., accumulated posteriors of each acoustic unit, HMM state transitions and GMM component) that are computed by the first-pass AUD model on raw features, for updating the model posterior distributions in the first iteration of second pass training; we find empirically, this procedure outperforms conducting the second-pass training on LDA features from scratch (i.e., initializing posterior distributions the same as their priors).

Using the Kaldi toolkit [45], we conduct our multilingual TDNN-based BN training with 10 language collections provided in the IARPA Babel Program (IARPA-BAA-11-02): Assamese, Bengali, Cantonese, Haitian, Lao, Pashto, Tamil, Tagalog, Vietnamese and Zulu. 10-hour transcribed speech of each language is used for training. We first evaluated our multilingual BN recipe for ASR experiments using this Babel corpus, and observed modest WER improvements in the hybrid multilingual TDNN system by using other languages to supplement the training data of test language, and more robust WER improvements in the tandem TDNN system with spliced multilingual TDNN-based BN features and MFCCs. Detailed discussion of ASR results is beyond the scope of this chapter. Particularly, we are interested in learning speech representations with effective cross-lingual generalization to an unseen language as in a zero-resource setting where AUD is typically performed. The multilingual TDNN-based BN training recipes will be available in the Kaldi code repository [45] as an open-source capability of language-independent BN feature extraction.

5.4.3 AUD Evaluations

For same-different tasks, from the time aligned word transcriptions, we extracted all word examples that are at least 0.50 s in duration and at least 6 characters as text from development and evaluation data set respectively. Development set produces approximately 11k word tokens, 60.8M word pairs of which 96.8k have the same word type. Evaluation data has 19k tokens and 186.9M word pairs with 281.8k pairs having the same word type. 200 ($T = 200$) dimensional AUD posteriorgram features across HMM units are produced as inputs to DTW scoring function.

For document classification, we use the acoustic unit trigram representation, and we scale each trigram feature value by the inverse document frequency, referred to as TFIDF features. We further normalize each feature vector to $\mathcal{L}^2$ norm unit length. To be comparable with experimental results in [35], classification accuracies are reported based on 10-fold cross validation, and average performance

with standard deviations reported in Table 5.1.

For document clustering, we use the TFIDF features of the spliced acoustic unit unigrams, bigrams and trigrams. Purity and B-Cubed F1 score [46, 47] are used as evaluation metrics. We run all clustering experiments using Cluto clustering library [43], and for each one, we use 10 different initializations and report average performance and standard deviations.

As shown in Table 5.1, for NMI, same-different task and document classification, both LDA and BN features produce substantial and correlated improvements, except that performing LDA on MFCCs does not seem to improve the same-different AP. Specifically, the best performance across all measures by combining LDA and BN demonstrates the complementarity between these two approaches. Given all the same AUD model configurations, we find the improved same-different AP or document classification accuracy often indicates the NMI improvement, which implies in a zero-resource setting, AUD evaluation can fall back to other resources if necessary, e.g., word pairs or topic labels, which might be easier to be available or to obtain than the expensive orthographic phoneme transcripts.

Also, as we can see, NMI only drops slightly between development and evaluation data, which shows the learned acoustic unit models can generalize well on unseen data.

Moreover, we find directly using the raw MFCCs after CMVN as acoustic features for all word segments gives AP 0.208 on the same-different task of development data. Therefore, the significantly higher AP 0.247 provided by our AUD posterior features across acoustic unit HMMs (learned from MFCCs) demonstrates AUD posteriorgrams as effective acoustic representations. We also find the 600-dimensional AUD posterior features across each acoustic unit HMM state can provide even higher AP as 0.267, and we leave all the applications of HMM-state based posterior features for the future work, since HMM state-level posteriors (with 3 times larger dimensions than HMM unit-level posteriors if we use 3 state HMM) have much larger computational overhead in DTW scoring.

Also, our legitimate zero-resource document classification effort with LDA- and BN-based AUD yields accuracy 0.73, which demonstrates AUD tokenizations to be effective document representations for discriminative tasks. For topic clustering in development data, there are consistently marginal gains as other measures improved. However, this trend does not well hold in evaluation data. Moreover, on the same development data we use, [35] shows phone trigram features by a high-resource supervised phoneme recognizer give classification accuracy up to 0.9138, clustering purity 0.6194 and B^3 F1 score 0.5256, which indicates document processing with unsupervised phonetic information remaining a challenging task.

5.5 Conclusions

We present an effective AUD framework can be successfully improved by integrating a self-supervised LDA technique and a complementary language-independent TDNN-based BN feature extraction recipe. We demonstrate the effectiveness of AUD-based discriminative features as acoustic representations given by AUD posteriors across automatically discovered units, and as document representations given by AUD tokenizations. Moreover, we find the gains in the intrinsic NMI metric for AUD algorithm development can often be predicted by the improved efficacy of applying AUD to real speech applications like same-different task and document classification. This suggests that in real zero-resource scenarios, as we optimize the core AUD technology, alternative evaluations by various different resources can be considered, which serve as zero resource efforts towards ASR technology without replying on any expert-provided linguistic knowledge.

Chapter 6

Topic identification of spoken documents using unsupervised acoustic unit discovery

Santosh Kesiraju, Raghavendra Pappagari, Lucas Ondel, Lukáš Burget, Najim Dehak, Sanjeev Khudanpur and Jan "Honza" Černocký, Suryakanth V Gangashetty

This chapter investigates the application of unsupervised acoustic unit discovery for topic identification (topic ID) of spoken audio documents. The acoustic unit discovery method is based on a non-parametric Bayesian phone-loop model that segments a speech utterance into phone-like categories. The discovered phone-like (acoustic) units are further fed into the conventional topic ID framework. Using multilingual bottle neck features for the acoustic unit discovery, we show that the proposed method outperforms other systems that are based on cross-lingual phoneme recognizer.

6.1 Introduction

Recent advances in machine learning and spoken language technologies have given rise to many daily life applications. This progress is mainly coming from the so called "deep learning" methods, that requires large amounts of labelled data for training. Unfortunately, for many languages the lack of labelled data preclude the direct application of state-of-art spoken language technologies.

The need for automatic analysis of spoken documents is important, since the amount of and the ability to store multimedia data is increasing day-by-day. The technologies developed in this regard are primarily useful for tasks such as query based document retrieval, topic identification (topic ID), key-word spotting, etc,. Most of these information retrieval tasks rely on the semantics in a document, where the notion of topics play an important role. One particular task of interest is topic ID, where the goal of a system is to identify the topics of the spoken documents in a given collection. This can also be seen as a supervised task, where a given document has to be classified into one of the pre-defined topics.

The majority of the systems for topic ID of spoken documents use word or phoneme based automatic speech recognition (ASR) as the pre-processing step, followed by the application of techniques developed by the text retrieval community [82, 83, 39]. It is possible to train ASR systems for English on large amounts (1000 hours) of publicly available data [84] and software [85]. But, not every language is rich in resources for building ASR systems, hence there is a need for developing techniques that are useful for languages with low or zero resources. In this chapter, we propose a topic ID system that relies on the unsupervised discovery of acoustic (phone-like) units using a non-parametric Bayesian

model.

Earlier work on the analysis of spoken documents in zero resource scenarios was based on identifying recurrent patterns of speech (spoken words), where dynamic time warping (DTW) based algorithms were used [86, 87]. However, these are not scalable to large amounts of data. An alternative is to use phone recognizers from other languages. This idea was explored for the task of topic ID in [88]. Under limited resource conditions, i. e., with limited vocabulary for training an ASR, topic ID of spoken documents was explored in [89].

We have recently proposed an infinite phone-loop model [90], similar to [6], to automatically segment unlabelled speech into phone-like categories. By using Variational Bayes rather than Gibbs sampling, we have shown that this model can be trained efficiently on large speech corpora with greater accuracy [90]. We use this model as a front-end to a topic ID system. A similar idea was proposed in [91, 92], where the authors used "self-organizing-units" to represent speech into meaningful tokens. In our work, we jointly learn the speech segmentation and the parameters of the acoustic model in a completely unsupervised fashion, whereas the earlier approaches [91, 6], learn the segmentation independently of the acoustic model. In [91], the acoustic model is learnt together with the language model, whereas we limit ourselves to model the acoustic data.

The infinite phone-loop model is described in Section 6.2, and our topic ID framework is explained in Section 6.3. Section 6.4 includes the details of the data set, description of the baseline and the proposed systems. We provide the results of topic ID systems in Section 6.5, followed by conclusions in Section 6.6.

6.2 The infinite phone-loop model

6.2.1 Model

The model aims at segmenting and clustering unlabelled speech data into phone-like categories. It is similar to a phone-loop model in which each phone-like unit is modelled by an HMM, and each HMM state distribution is represented by a GMM. This phone-loop model is fully Bayesian in the sense that:

- it incorporates prior distributions over HMM state transition probabilities, and parameters of state emission GMM distributions,
- it has a prior distribution over the units modelled by a Dirichlet process [10].

Informally, the Dirichlet process prior can be seen as a standard Dirichlet distribution prior for a Bayesian mixture with an infinite number of components. However, we assume that our N data samples have been generated with only M components ($M \leq N$) from the infinite mixture. Hence, the model is no longer restricted to have a fixed number of components but instead can learn its complexity (i. e. number of components used, M) according to the training data. The generation of a data set with M speech units can be summarized as follows:

1. sample the vector $\mathbf{v} = v_1, ..., v_M$ with

$$v_i \sim \text{Beta}(1, \gamma) \tag{6.1}$$

   where γ is the concentration parameter of the Dirichlet process

2. sample parameters of M HMMs, θ_1,...,θ_M from the prior (base) distribution of the Dirichlet process.

3. sample each segment as follows:

(a) choose a HMM parameters with probability $\pi_i(\mathbf{v})$ (using *stick breaking process* [9]) defined as:

$$\pi_i(\mathbf{v}) = v_i \prod_{j=1}^{i-1}(1 - v_j) \tag{6.2}$$

(b) sample a path $\mathbf{s} = s_1, ..., s_n$ from the HMM transition probability distribution

(c) for each s_i in $\mathbf{s}$:

i. choose a Gaussian components from the mixture model

ii. sample a data point from the Gaussian density function

6.2.2 Model parameters

In the absence of information about the prior distribution of the parameters of the model, it is convenient to use conjugate prior (distribution), which greatly simplifies the inversion of the model: indeed, due to the conjugacy, the posterior distribution of each parameter of the model will have the same parametric form of the prior. The distribution of the mean $\boldsymbol{\mu}$ and the diagonal covariance matrix $\boldsymbol{\Sigma}$ with diagonal $\boldsymbol{\lambda}$ is modelled by a Normal-Gamma density: $\mathcal{N}(\boldsymbol{\mu}|\boldsymbol{\mu_0}, (\kappa_0\boldsymbol{\lambda})^{-1})$ Gamma$(\boldsymbol{\lambda}|\alpha_0, \boldsymbol{\beta_0})$ where $\boldsymbol{\beta}_0$ is the rate parameter of the Gamma distribution. The prior of the weights $\boldsymbol{\pi}$ of a GMM and the row r of the transition matrix of an HMM are modelled by Dirichlet distributions parametrized by the vectors $\boldsymbol{\eta}_0^{(gmm)}$ and $\boldsymbol{\eta}_0^{(hmm,r)}$ respectively. Finally, the prior distribution over the proportions v_i is the Beta$(1, \gamma)$ distribution. The model has also 3 set of hidden variables:

- c_i the index of the HMM for the i^{th} segment in the data set
- s_{ij} the HMM state of the j^{th} frame in the i^{th} segment
- m_{ij} the GMM component of the j^{th} frame in the i^{th} segment.

6.2.3 Inference

We would like to invert the model previously defined to obtain the probability of the parameters, and the hidden variables which define the segmentation, given the data. Following variational Bayes (VB) framework, it can be achieved by optimizing a lower-bound on the log-evidence of the data with respect to the distribution over the parameters q:

$$\begin{aligned}\log p(X) \geq & E_q[\log p(\mathbf{X}, \mathbf{c}, \mathbf{S}, \mathbf{M}, \boldsymbol{\Theta}|\boldsymbol{\Phi}_0))] \\ & - E_q[\log q(\mathbf{c}, \mathbf{S}, \mathbf{M}, \boldsymbol{\Theta})]\end{aligned} \tag{6.3}$$

where $\mathbf{X}$ is the entire set of features of the N segments, $\mathbf{c} = c_1, ..., c_N$, $\mathbf{S} = s_{11}, ..., s_{NL_N}$, $\mathbf{M} = m_{11}, ..., m_{NL_N}$, $\boldsymbol{\Theta}$ is the set of all the parameters and $\boldsymbol{\Phi}_0$ is the set of the hyper-parameters of the prior distribution over the parameters. The equality is achieved if and only if $q(\mathbf{c}, \mathbf{S}, \mathbf{M}, \boldsymbol{\Theta}) = p(\mathbf{c}, \mathbf{S}, \mathbf{M}, \boldsymbol{\Theta} \mid \mathbf{X})$. Because of the conjugate prior distribution described in Section 6.2.2, we have a closed form solution [9] for a co-ordinate ascent algorithm, when considering the mean-field approximation:

$$q(\mathbf{c}, \mathbf{S}, \mathbf{M}, \boldsymbol{\Theta}) = q(\mathbf{c}, \mathbf{S}, \mathbf{M})q(\boldsymbol{\Theta}), \tag{6.4}$$

where we have assumed the statistical independence between the parameters and the hidden variables of the model. Following [9], another approximation is done to cope with the infinite number of components in the mixture; we set $v_T = 1$ to force the weight of any component greater than T to zero. By using the factorization in (6.4) and variational calculus, one can show that the (log) distributions that maximizes the bound (6.3) are :

$$\begin{aligned} \log q^*(\mathbf{c}, \mathbf{S}, \mathbf{M}) &= E_{q(\mathbf{\Theta})}[\log p(\mathbf{X}, \mathbf{c}, \mathbf{S}, \mathbf{M}, \mathbf{\Theta}|\mathbf{\Phi}_0)] + \text{const} \\ \log q^*(\mathbf{\Theta}) &= E_{q(\mathbf{c},\mathbf{S},\mathbf{M})}[\log p(\mathbf{X}, \mathbf{c}, \mathbf{S}, \mathbf{M}, \mathbf{\Theta}|\mathbf{\Phi}_0)] + \text{const} \end{aligned} \tag{6.5}$$

Maximizing the bound (6.3) minimizes the KL divergence between (6.4) and the true posterior distribution of model parameters. Therefore (6.4) can be taken as the approximate posterior, which is found by evaluating each factor in turn using (6.5) until convergence. Details about the update equations can be found in [90].

The mixture of HMMs can be interpreted as a single compound HMM, which allows us to easily evaluate the approximate posterior distribution $q(\mathbf{c}, \mathbf{S}, \mathbf{M})$ using the standard forward-backward (Baum-Welch) algorithm. Similarly, Viterbi algorithm can be used for decoding the sequences of the discovered acoustic units. This subtlety simplifies the inference algorithm as we do not need any pre-segmentation of the speech data.

6.3 Topic ID framework

6.3.1 Topic ID in low resource scenarios

Let D be the collection of documents comprising a vocabulary V, and let each document belong to one and only one topic from a set of T topics. Let d, w and t be the variables for denoting documents, tokens in the vocabulary and topics respectively. Assuming the *bag-of-words* approach, each spoken document d is represented in the form of a vector, whose dimension is equal to the size of the vocabulary V. In the conventional topic ID framework, the vocabulary V is simply the set of words as seen in the document collection. In low resource scenarios, when a reliable word based ASR is not available, the vocabulary can be made from phoneme n-grams (usually $n = 3, 4$). It was observed that the topic ID based on phoneme trigrams is a robust alternative to a word based topic ID system [83]. Since the infinite phone-loop model discovers phone-like units, we experimented with 3-grams and 4-grams as the terms (word-types) in the vocabulary.

6.3.2 Vocabulary selection

In a supervised setting, vocabulary selection plays an important role as it can drastically reduce the dimension of the document vectors and significantly improve the performance of the classifier. The n-grams for vocabulary are chosen based on conditional probabilities as proposed in [83]. The conditional probability of topic t given a n-gram w is estimated as follows:

$$P(t \mid w) = \frac{f_{wt} + |T|\, P(t)}{f_w + |T|}, \tag{6.6}$$

where f_{wt} is the number of times the n-gram w appeared in documents related to topic t, f_w is the total number of times n-gram w appeared in all the documents from the training set. $P(t)$ is the probability of topic t as estimated from training corpus. The conditional probability in (6.6) is computed for every topic and the vocabulary is formed by considering top N_t n-grams per topic with the highest probabilities (6.6).

6.3.3 Document representation

If f_{wd} represents the frequency of token w in document d, then the smoothed TF-IDF (term frequency - inverse document frequency) representation (v_{wd}) is given by,

$$v_{wd} = f_{wd} \,.\, \log\Big(\frac{|D|}{1 + N_{dw}}\Big) + 1, \tag{6.7}$$

where N_{dw} represents the number of documents in which the term w appears. The resulting document vectors are further ℓ_2 normalized, such that the sum of the squares of elements equals to 1.

Table 6.1: *The statistics show number of recordings per topic from a subset of Fisher corpus used in the preliminary experiments.*

Topic Name	**# docs.**	
	Training set	**Test set**
Anonymous Benefactor	20	56
Corporate Conduct in the US	20	38
Education	20	57
Holidays	20	58
Illness	20	71
Minimum Wage	20	144
Total duration (hrs).	21.67	77.28

6.3.4 Document classification

For classifying the documents, we have used linear support vector machines, trained using stochastic gradient descent [93, 42]. The SVMs are used in a one-versus-all strategy for multi-class classification. On the training data, we used 5-fold cross validation and performed grid search over the choice of hyper-parameters (i. e., choice of ℓ_1, ℓ_2, elastic net regularization and the regularization coefficient) of the classifier. Using the best of hyper-parameters, the classifier is trained again using all the training data to predict the topic labels of the test documents.

6.4 Experimental setup

6.4.1 Data set

Our experiments on topic ID are conducted on the Fisher phase 1 English corpus, which is a collection of recordings from conversational telephone speech. Each document represents one telephone conversation that includes both sides of the call, and is associated to one and only one topic. We chose a *subset* that consists of the same 6 topics as in [87], but relatively more number of documents per topic. The details of this subset of data used in our experiments is given in Table 6.1. This subset was chosen to study the acoustic unit discovery (AUD) model. We have also experimented on a larger set of 40 topics with the same data splits as used in [83, 39, 88].

6.4.2 Oracle system

The oracle system is based on the English phoneme recognizer trained on Fisher corpus with large amounts ($\sim$ 500 hrs.) of data. The motivation for using such a setup is to show the performance of a topic ID system in scenarios where the target language is known and considerably large amounts of training data is also available. We used DNN based phoneme recognizer built with the Kaldi toolkit following the recipe described in [94].

6.4.3 Baseline systems

The baseline systems are based on phoneme recognizers from various languages: Czech, Hungarian, Russian, which were trained with split temporal context features [95]; and Turkish, from the Babel program, which was trained in a similar framework as described in [96]. The Hungarian phone recognizer was used as a baseline comparison for the task of topic ID in [83, 88, 91, 92].

6.4.4 Proposed system

The proposed system is based on the discovered acoustic units from the infinite phone-loop model. We explored the following set of input speech features for training the model:

1. 13 dimensional MFCCs + Δ + $\Delta\Delta$
2. Multilingual bottle neck features (Babel-MBN) [97].
3. Multilingual bottle neck features (global phone dataset, GP-MBN) [98].

The Babel-MBN features are extracted using bottle neck neural network trained on data comprising of Cantonese, Pashto, Tagalog, Turkish and Vietnamese and GP-MBN are trained on data comprising of Czech, German, Portuguese, Russian, Spanish, Turkish and Vietnamese languages. Both the neural networks are trained in the same fashion as described in [97].

The hyper-parameters of the infinite phone-loop model play a significant role in quality and quantity of the discovered acoustic units. We primarily experimented with the concentration (γ) of the Dirichlet process prior and the truncation (M). The effect of these hyper-parameters is explained in the following section along with the results. The rest of the hyper-parameters i. e., states per HMM ($\mathcal{S} = 3$) and Gaussian components per state ($\mathcal{C} = 2$) are fixed. We also investigate the importance on the amount of data used to train the infinite phone-loop models.

6.5 Results

In the first section of the results, we give the comparison of topic ID systems across various baselines and AUD systems. All the systems are based on 1-best sequence from the recognizers. These experiments are performed on a *subset* of 6 topics from the corpus as detailed in Table 6.1. In the later section, we show the topic ID results on a *larger* set of 40 topics from the same corpus.

6.5.1 Topic ID on the subset

The AUD model was trained on the 21 hr. training set as presented in Table 6.1, and the trained model was used to automatically transcribe both the training and test data in terms of the discovered acoustic units. The resulting automatic transcription was fed into the topic ID framework that was described in Section 6.3. Here, both the AUD and topic ID models are trained on the same 21 hr. training set (Table 6.1).

The classification accuracy (in %) of the topic ID systems based on various phoneme recognizers (baseline and oracle) and the discovered acoustic units (AUD) are presented in Table 6.2. The proposed infinite phone-loop model outperforms all other phone recognizers except for the one trained on the English (target language). This shows that systems trained on another phone set than the target one are far from being optimal, and it is preferable to use unsupervised methods instead. The vocabulary size (set of all unique trigrams) of the proposed system is however much bigger than baseline systems, as the number of discovered acoustic units is 100 (which is larger than the number phoneme set of the other phone recognizers). In Table 6.2, the results are reported only for the vocabulary size for which the classification accuracy is observed to be highest.

Topic ID across various AUD systems

This section presents the comparison of several AUD systems that were explained in Section 6.4.4. We primarily experimented with various types of input speech features and concentration (γ) parameter of the Dirichlet process. Higher concentration ($\gamma > 1$) encourages more number of clusters (i. e., in the stick-breaking process, higher concentration results in more number of smaller chunks of the stick).

Table 6.2: *Comparison of Topic ID accuracy (in %) on the subset of 6 topics across various systems for the best set of* 3*-gram vocabulary.*

Recognizer	**Acc. (%)**	**Vocabulary size ($\|V\|$)**
Hungarian (HU)	70.19	2428
Czech (CZ)	67.36	5856
Russian (RU)	60.90	3027
Turkish (TU)	55.04	12041
Proposed (AUD)	**76.48**	3029
Oracle (EN)	98.96	9516

From Table 6.3, we can observe that multilingual bottle-neck features are a better representation of speech for unsupervised learning of acoustic units, and therefore results in better topic ID accuracy.

Table 6.3: *Comparison of Topic ID accuracy (in %) on the subset of 6 topics across various AUD systems.*

Feature type	**Accuracy**	
	$\gamma = 1.0$	$\gamma = 10.0$
MFCC	36.33	39.27
Babel-MBN	63.41	75.47
GP-MBN	72.74	**76.48**

6.5.2 Topic ID on the large set

The details of the topic ID training and test splits on a large set of 40 topics from Fisher corpus are presented in Table 6.4. These are the same splits as used in [83, 39, 88]. For these experiments, we have trained two AUD models, one with 26 hrs. (AUD-26) and the other with 52 hrs. (AUD-52), and neither of them overlap with any of the topic ID training or test data from Table 6.4. These two AUD models are trained with concentration, $\gamma = 10$ and GP-MBN input speech feature representation, as this combination was observed to be giving the best topic ID performance earlier (Table 6.3). After the AUD models are trained, they are used transcribe the topic ID training and test data (Table 6.4) in terms of the discovered acoustic units, followed by the topic ID framework described earlier in Section 6.3.

We chose the best baseline system (i. e., Hungarian, HU) from Table 6.2 and perform the topic ID experiments on this large set of 40 topics in the same framework. All these results are presented in Table 6.5, and we can observe that the proposed AUD systems are better than the baseline, but still far from the oracle system (DNN based English phoneme recognizer). This is partly because we have a more difficult task of classifying 40 topics.

From these experiments, we observe that in an unknown scenario and/or language, it is better to borrow knowledge from the other languages at a lower (feature) level (multi-lingual bottle neck features) than at a much higher level (phone recognizer) and rely on the unsupervised methods to discover the acoustic units from the data and use them for further tasks.

Table 6.4: *Statistics of the data splits from large set of Fisher phase 1 corpus used in the experiments.*

Set	**# docs.**	**Duration (hrs.)**	**# topics**
Topic ID training	1374	244	40
Topic ID test	1372	226	40

Table 6.5: *Comparison of Topic ID accuracy (in %) on the large set of 40 topics for the best set of n-grams from the vocabulary.*

Recognizer	**Acc. (%)**	$\|V\|$	n**-gram**	**AUD params.**
AUD-26	**53.84**	6061	3	$M = 200, \gamma = 10$
AUD-52	**55.54**	2140	4	$M = 100, \gamma = 10$
HU	47.92	25351	3	-
EN	91.41	11236	3	-

6.6 Conclusions

This work focuses on the importance and application of unsupervised acoustic unit discovery for the task of topic identification. We showed that using multilingual bottle-neck features for learning the acoustic units, the performance of the topic ID system could be improved significantly. Our experiments on a corpus of conversational telephone speech showed that the proposed system performs better than the other systems which rely on the cross-lingual phoneme recognizers. Although the results are encouraging, there is still a significant space for improvement to reach the performance of supervised speech recognition systems. One step towards achieving this would be to jointly learn the language model and the infinite phone-loop parameters in an unsupervised fashion.

Chapter 7

Combining Acoustic and Lexical Unit Discovery Towards Unsupervised LVCSR

Thomas Glarner, Oliver Walter, Reinhold Haeb-Umbach

We present a hierarchical approach to training a speech recognition system from untranscribed speech. The system consists of modules for the unsupervised discovery of both, acoustic and lexical units, where the first correspond to phonemes and the latter to words. While solutions for either of the two tasks can be found in the literature, the purpose of this contribution is to couple the two to learn a hierarchical representation of speech with acoustic units on the lower and word-like units on the upper level of the hierarchy. Nonparametric Bayesian models are employed for both tasks to cope with the a priori unknown number of models. Further, a feedback loop from the word discovery unit to the acoustic model discovery unit is proposed to improve the latter by exploiting the language model information learned in the former. Initial experiments on the Xitsonga language show that this feedback indeed improves the quality of the discovered acoustic units.

7.1 Introduction

Transcription costs make annotated corpora expensive to create, and linguistic expert knowledge is required to compile pronunciation dictionaries. In contrast to this, raw audio data is cheap to obtain. Furthermore, there are many languages which are only spoken by a small number of people, rendering the creation of annotated corpora uneconomical.

However, building a speech recognizer from audio only is a widely unsolved challenge, which calls for unsupervised learning techniques. Bayesian techniques are of particular interest since they allow for the incorporation of prior knowledge (e.g., the Zipf law in the case of language modeling), and because they can express uncertainty in a formal way, which helps to avoid premature decisions on tokens before exploiting all available knowledge sources. Furthermore, in the case of acoustic unit (AU) and word discovery (WD), the number of AUs and words are not known beforehand, which can be approached with nonparametric Bayesian techniques.

In this contribution, our working hypothesis is that the hierarchical structure of speech, where the words are composed of elementary AUs, the phonemes, should also be reflected in the unsupervised learning approach, which should attempt to learn this hierarchical representation. We thus need to solve two subtasks, acoustic unit discovery (AUD) and WD on top of the AUD. While either of the two tasks has been tackled earlier in isolation, this chapter aims at combining the two. This contrasts with approaches which directly model word-like units as in [48], which we believe will become more

difficult as the vocabulary size grows.

While there have been many approaches to learn the elementary acoustic units from data, more recently Bayesian approaches have become prevalent. In [49] a nonparametric Bayesian model has been proposed which later has been extended to jointly learn the acoustic units and the segment boundaries [50]. While these two approaches relied on Gibbs sampling, the variational Bayesian approach of [51] is computationally less expensive while achieving comparable, if not better recognition performance.

The subtask of WD has mostly been studied on text input, where it corresponds to inserting white spaces in a character stream, i.e., carrying out a segmentation task. Again, Bayesian methods have been shown to be quite effective. The work of [52] employs a Hierarchical Dirichlet Process (DP) for word segmentation, while [53] introduces a Nested Hierarchical Pitman-Yor Process, consisting of hierarchical Bayesian language models at the word and the character level. Word discovery is achieved by iterating between word segmentation, given a language model, and language model estimation for a given word segmentation.

Only few works have considered the segmentation of a label sequence produced by an ASR phoneme recognizer. This is a much harder task, since the label sequence contains recognition errors. A system for the segmentation of input phoneme lattices based on Weighted Finite State Transducers has been proposed in [54]. In own prior work we have shown that unsupervised word segmentation on phoneme lattices produced by an ASR decoder can even improve the ASR decoder result [55, 56]: Using the language model learned jointly with the word segmentation in the next iteration of the ASR decoder led to an improved phoneme recognition rate. This observation was the motivation for the work described here.

In this contribution, we combine the nonparametric Bayesian AU discovery from [51] with the word discovery and segmentation system proposed in [55, 56]. This results in a large vocabulary subword-unit based speech recognizer, which is trained in a completely unsupervised manner. Due to the lack of supervision, both acoustic unit and word discovery have high error rates. In an attempt to cope with this issue we employ lattices as interface between the two, such that the word discovery is able to correct errors which may be present in the first-best label sequence produced by the AU discovery. Further, we propose a feedback loop architecture, where the best-scoring label sequence, according to the word discovery module, is used as tentative transcription to retrain the AU discovery system.

We evaluated the proposed system on the Xitsonga corpus which has been provided for a recent zero resource challenge [57]. While the observed improvements in AU discovery by the feedback loop are modest, we nevertheless believe that the proposed hierarchical approach is promising for future work.

7.2 Modules of the unsupervised ASR system

The proposed feedback system has its foundation in two components that tackle different subtasks of unsupervised speech recognition: An AUD component and a WD component. The interface between the components is given through a lattice of acoustic units. The following sections give a brief overview of these components.

7.2.1 Acoustic unit discovery

The AUD system component is the one proposed by Ondel, Burget and Cernocky [51]. It is similar to the approach proposed by Lee et al. [49] in that the acoustic model consists of an infinite mixture of HMMs based on a DP prior. However, they differ in the way they treat the DP: In the work of Lee et al., the Chinese Restaurant Process representation is used to perform inference by Gibbs Sampling. In contrast to this, Ondel et al. employ a variational approximation based on a truncated Stick Breaking Process analogous to [58]. Here, the main idea is that the true distribution is modeled

as a full Dirichlet Process and thus an infinite HMM mixture model. A variational approximation is stated under the assumption that, after a fixed truncation length, subsequent mixture units can be neglected due to their vanishing probability. The learning algorithm aims to maximize the similarity between the truncated mixture and the original Dirichlet Process. The variational approach allows for fast and easily parallelizable Bayesian inference.

Furthermore, no explicit boundary variable between the acoustic units is included in the model. Instead, the sequence of acoustic units within an utterance is modeled by a phone loop: Any acoustic unit HMMs can follow any other, and when the end state of one HMM is reached, the next HMM is chosen according to its unit probability given under the DP.

The acoustic unit discovery training is an Expectation-Maximization type of algorithm, which iterates between the E-step (modified forward-backward algorithm to obtain state posteriors) and the M-step (update of the model parameters), see [51] for details.

A drawback is that the variational inference can handle only unigram acoustic unit probabilities. For higher-order models, such as a bigram, no variational inference algorithm is known [58]. To overcome this drawback, we propose to incorporate longer-range context information by feeding back the result of the word discovery module as described later on.

7.2.2 Word discovery

The word discovery or word segmentation module is the one proposed in [56]. It is based on the system in [54], which in turn relies on the language model introduced by [53].

The underlying assumption is that the sequence of acoustic units within a word can be better predicted than at word boundaries. The predictive probabilities are given by a Nested Hierarchical Pitman-Yor language model, which is estimated alongside the word segmentation task. It comprises two hierarchical Pitman-Yor language models (HPYLMs), one at the word and one at the AU level [59]. The predictive probability of a word w appearing in a context $\mathbf{u}$ is given by:

$$\Pr(w|\mathbf{u}) = \frac{c_{\mathbf{u}w\cdot} - d_{|\mathbf{u}|}t_{\mathbf{u}w}}{\theta_{|\mathbf{u}|} + c_{\mathbf{u}\cdot\cdot}} + \frac{\theta_{|\mathbf{u}|} + d_{|\mathbf{u}|}t_{\mathbf{u}\cdot}}{\theta_{|\mathbf{u}|} + c_{\mathbf{u}\cdot\cdot}}\mathbf{Pr}(w|\pi(\mathbf{u})). \tag{7.1}$$

If $\Pr(w|\mathbf{u})$ is an n-gram probability, $\mathbf{u}$ denotes the context of length $n-1$. $c_{\mathbf{u}w\cdot}$ is the number of times the word w has been seen in the context $\mathbf{u}$, and $\pi(\mathbf{u})$ denotes the shortened context of length $n-2$. The count $t_{\mathbf{u}w}$ stands for the number of distinct draws from the base measure (the prior probability) for the given context (the so-called number of tables in the Chinese Restaurant Process). The parameters of the model are the concentration $\theta_{|\mathbf{u}|}$ and the discount parameter $d_{|\mathbf{u}|}$. While the first controls the variation of the distribution around its base measure, the second parameter is what differentiates the Pitman-Yor process from the DP: it makes sure that the resulting distribution follows Zipf's law. In (7.1), the dot stands for any character. E.g., $c_{\mathbf{u}w\cdot}$ stands for number of occurrences of the word w in the context $\mathbf{u}$, irrespective of the "table" it is assigned to. For more details see [59].

At the root of the HPYLM is the zerogram language model. However, since the vocabulary and thus its size is unknown, the zerogram probability, which is equal to one over the vocabulary size, cannot be computed. Instead, the probability of the AU sequence that the word consists of is used. This probability is calculated using another HPYLM over the AUs. This is the nested HPYLM, which allows for a potentially infinite vocabulary size [53]. The entire language model is called segmentation LM in the remainder of the chapter.

To learn new words, a (blocked) Gibbs sampling scheme is performed. Given a language model, a new segmentation of an utterance is sampled using the forward filtering backward sampling algorithm [53]. With this segmentation the language model probabilities can be updated. Then a new utterance is chosen and the scheme is repeated until convergence.

Furthermore, since the AU sequence provided by the AUD module to the WD module contains recognition errors, the interface between the two modules is realized by a lattice, which contains the

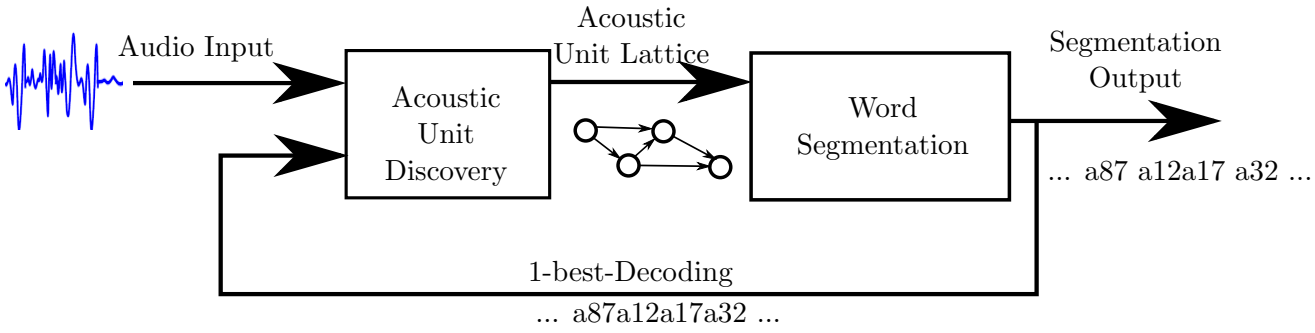


Figure 7.1: Complete AUD+WS system including the proposed feedback loop.

set of the most likely sequences of acoustic units in a directed acyclic graph structure. Since the lattice may contain sequences with fewer errors than in the 1-best path, the word segmentation is able to correct errors by choosing an alternative to the sequence considered most probable by the AUD module. This selection is done by rescoring the phoneme lattice with an additional rescoring AU HPYLM learned in the WD module.

As explained in [56], this rescoring AU HPYLM needs to be different from the AU part in the segmentation LM for two reasons: Firstly, the AU level part of the segmentation LM is only trained with a fraction of the corpus, namely those AU sequences which cannot be identified to be part of a word. Secondly, both language model orders should be chosen differently, since the rescoring LM seems to benefit from a high AU LM order while the word segmentation LM does not, probably due to the different training set sizes. Further, this second LM includes a word end tag and calculates the word end probability at the AU level and therefore incorporates the knowledge of the word level part.

7.2.3 Word-level information feedback

In order to improve the result of the acoustic unit discovery, this work proposes to exploit the output of the word segmentation component. Both the rescored lattice and the segmentation output comprise valuable information about the likelihood of acoustic unit sequences over a long-range context. This information is fed back by extracting the best path from the rescored lattice. The corresponding AU sequence serves as the transcription for a forced alignment in the Viterbi step of the AUD component. With this initialization the AU training is repeated to improve the AU discovery. The full system is shown in Fig. 7.1.

7.3 Experiments

The proposed system is evaluated on the Xitsonga corpus of the Zero Resource challenge [60]. The corpus consists of read speech from 24 different speakers totaling about 2 hours and 30 minutes of speech with a vocabulary of 2288 words.

Standard MFCC feature vectors are extracted from the audio input. The concentration parameter of the DP for modeling the acoustic unit probabilities in the AUD component is set to 1, and the stick breaking process is truncated at 100 acoustic units. For each acoustic unit, a three-state-HMM in left-to-right topology is assumed with two-component Gaussian mixture models as emission distributions. The AUD training is performed with 20 iterations, where each iteration consists of the estimation of the latent variables (E-step) and estimation of the model posterior parameters (M-step).

For the segmentation LM, the word LM is always a unigram model since we observed that the acoustic units are too noisy to obtain robust n-grams of higher order. The orders of the AU LMs – the unit LM part of the segmentation LM and the rescoring LM – were set to higher values, though.

After the word segmentation stage, the best AU sequence is fed back to the AUD stage, and 10 additional iterations of the AUD training are carried out.

Performance of the AUD module is measured in terms of the normalized mutual information between the discovered units and the ground truth: Given the labels Y obtained by labeling the evaluation subset with the trained model and the corresponding true labels X from the transcriptions, a confusion matrix is calculated and the mutual information is obtained by

$$I(X;Y) = H(X) - H(X|Y) = \mathrm{E}\left[\log\left(\frac{\Pr(X|Y)}{\Pr(X)}\right)\right]. \tag{7.2}$$

It is convenient to normalize the mutual information with the phoneme label entropy $H(X)$, resulting in a measure between 0 and 1,

$$\mathrm{NMI} = \frac{I(X;Y)}{H(X)}, \tag{7.3}$$

where larger numbers indicate better performance.

The second performance measure is the ABX error between phonemic minimal pairs [61]. For example, the ABX discriminability between the minimal pair 'had' and 'hat' is defined as the probability that A and X are closer than B and X, where A and X are tokens of 'had', and B a token of 'hat' (or vice versa). As the distance measure the Kullback-Leibler divergence is used between frame-wise unit posteriorgram vectors.

Table 7.1 presents the results on the Xitsonga corpus for different LM orders for the AU LM part of the NHPYLM and the rescoring LM. All setups improve the measures of AUD performance compared to their initial values, with the best configuration improving the ABX error from 16.9 to 16.6%. These values compare favorably with the results obtained by other methods on the same data [60]. The NMI score is improved from 30.8% to 32.2%. While the feedback provides a consistent improvement of the performance measures, the actual LM orders do not seem to have a significant impact, as the achieved values are similar over a range of LM orders. A maximum token F-score of 2.5% and a type F-Score of 4.2% for the segmentation result and lexicon is obtained with a segmentation AU LM of order 2 and a rescoring LM of order 4, which is comparable to other results on the ZeroSpeech challenge as well.

Although the improvements obtained by feeding back the word discovery results to the AUD module are modest, they show that indeed AUD performance can be improved by incorporating long-range context information provided by the language model estimated in the word discovery module.

7.4 Conclusions

This chapter has presented a hierarchical approach to training an unsupervised speech recognition system with an acoustic unit discovery component and a word discovery component, which acts on the label sequence provided by the acoustic unit discovery. Representing word models as a composition of acoustic units makes this approach suitable for large vocabulary tasks. The system employs iterative algorithms at various levels: acoustic unit discovery is achieved by iterating between the estimation of the latent variables and latent parameters of the model; word discovery is done by iterating between word segmentation and language model estimation; and, finally, the acoustic unit discovery is improved by feeding back language model information estimated in the word discovery module. Experiments on the Xitsonga corpus of the Zero Resource challenge demonstrated the feasibility of

Table 7.1: Results for the Xitsonga corpus for different LM orders

Setup		NMI [%]	ABX error [%]
Initial AUD	Result	30.8	16.9
AU LM	Rescoring LM	NMI [%]	ABX error [%]
2	2	**32.24**	16.74
2	4	31.98	16.72
2	6	32.06	16.77
4	2	32.23	16.79
4	4	31.99	16.76
4	6	31.97	16.73
6	2	32.22	16.75
6	4	32.00	16.73
6	6	31.98	**16.64**

the approach. However, unsupervised large vocabulary ASR is still very much inferior to supervised ASR, demonstrating the difficulty of the task and calling for further research.

Chapter 8

Bayesian joint-sequence models for grapheme-to-phoneme conversion

Mirko Hannemann, Jan Trmal, Lucas Ondel, Santosh Kesiraju, and Lukáš Burget

We describe a fully Bayesian approach to grapheme-to-phoneme conversion. Similar to the joint-sequence models, we use a language model on graphone units (joint grapheme-phoneme pairs). However, we take a Bayesian approach using hierarchical Pitman-Yor-Processes. This provides an elegant alternative to using smoothing techniques to avoid over-training. No held-out sets and complex parameter tuning is necessary, and several convergence problems encountered in the discounted Expectation-Maximization (as used in the joint-sequence models) are avoided. Every step is modeled by weighted finite state transducers and implemented with standard operations from the OpenFST toolkit. We evaluate our model on a standard data set (CMUdict) and show that it gives comparable results to joint-sequence models in terms of phoneme-error rate while requiring a much smaller training/testing time. The most important advantage is that our model can be used in a Bayesian framework and for (partly) un-supervised training.

8.1 Introduction

Grapheme-to-phoneme conversion (G2P) refers to the task of converting a word from its orthographic form (sequence of letters / characters / graphemes) to its pronunciation (sequence of phonemes or other types of acoustic units). G2P has its application in speech synthesis and speech recognition. However, the techniques used for G2P can be applied to any monotonous translation problem.

To avoid the effort of manual rule crafting and to be able to generalize, most of the recent approaches to G2P are data-driven and probabilistic [62]. Recent discriminative approaches to G2P (e.g. [63]) seem to slightly outperform the generative ones. Both face the problem of over-fitting to the training data, which is alleviated by smoothing (e.g. [62]) and regularization techniques (e.g. [63]). The measurement of the training progress and the tuning of the smoothing/regularization parameters is done with the help of a held-out set. Smoothing and regularization changes the objective function of the training. Therefore, when tuning the parameters on the held-out set, the value of the objective function on the training set might deteriorate, or the training might even fail to converge. Since Bayesian methods have a notion of uncertainty of the model parameters, the model cannot over-train and no held-out set is necessary, i.e. all data can be used to estimate the model parameters. Our motivation is to design a model, that can be applied in a bigger Bayesian framework, e.g. in a Bayesian open-vocabulary speech recognizer. We also want to be able to deal with (partly) un-annotated data. For example, imagine that only a small root dictionary contains both orthographic form and pronun-

ciation, but a much bigger word list without pronunciations is available (from a text corpus), as well as a set of phoneme sequences without orthographic form that were recognized in places of out-of-vocabulary words. Therefore, we need a generative model that is also reversible, i.e. can be applied to the G2P and P2G task.

8.2 Relation to prior work

Many techniques have been proposed for the G2P problem [62]. Popular are joint-sequence models [62] and the publicly available tool Sequitur, which serves as our baseline. More recent work builds mainly on discriminative approaches: e.g. [64, 63, 65, 66, 67]. However, for the Bayesian approach, generative techniques are needed. Within a framework for unsupervised acoustic unit discovery, Lee et. al. [68] jointly learns a Bayesian model for G2P. Similar to our implementation, the training uses blocked Gibbs sampling of the letter-phoneme alignment to estimate the model parameters. However, [68] use a different parametrization (based on a context window around the current letter) and apply more restrictive constraints on the possible alignments (one letter can generate 0/1/2 phones). More importantly, the use of graphone units in our case makes the model reversible, i.e. the same model can be applied for G2P and P2G.

Similar to this work, Phonetisaurus [69, 70, 71, 72] (referring to [73]) also realizes G2P with the help of WFSTs and the OpenFST toolkit. Wu et. al. [74] use Phonetisaurus and OpenFST and incorporate conditional random fields and system combination. Phonetisaurus performs the segmentation (graphone alignment) as a separate step. The set of graphones is estimated using a context-less model (as an approximation to speed-up), and then a standard n-gram language model (LM) is estimated on the segmented training set. However, in our case, similar to [62], we jointly estimate the segmentation and the graphone LM. As opposed to [62], we do not use bottom-up model construction (step-wise 'ramping-up' and training LMs of increasing order). This approximation is not necessary in the Bayesian approach, we can immediately train the full order LM.

8.3 Joint-sequence models

Joint-sequence models [62] use a sequence of joint grapheme-phoneme units (graphones) to generate the orthographic form (letter sequence $\mathbf{g} \in G^*$) and pronunciation (phoneme sequence $\boldsymbol{\varphi} \in \Phi^*$) of a word. A graphone q is a pair of a letter sequence and a phoneme sequence of possibly different length and represents a mapping of $0..n$ letters to $0..m$ phonemes. The graphone inventory $\mathcal{Q}$ is usually derived automatically from the dataset:

$$q = (\mathbf{g_q}, \boldsymbol{\varphi_q}) \in \mathcal{Q} \subseteq G^* \times \Phi^*. \tag{8.1}$$

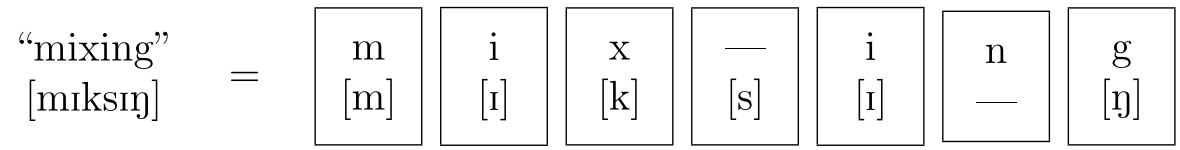


Figure 8.1: Graphone alignment for the word 'mixing' is a co-segmentation of spelling and pronunciation. Shown here using FST-style (01-to-01) graphones [62]: 0..1 letters map to 0..1 phonemes.

The spelling and the pronunciation are segmented into graphones using a co-segmentation (Fig. 8.1): the letter sequence $\mathbf{g}$ and the phoneme sequence $\boldsymbol{\varphi}$ are grouped into an equal number of segments K. For a given pair of letter and phoneme sequence, the segmentation into graphones is usally not unique. The task of graphone segmentation is to find (all) possible graphone sequences and

to calculate their probabilities. S is the set of all possible co-segmentations of $\mathbf{g}$ and $\boldsymbol{\varphi}$ (i.e. graphone sequences $\mathbf{q} \in \mathcal{Q}^*$):

$$S(\mathbf{g}, \boldsymbol{\varphi}) := \left\{ \mathbf{q} \in \mathcal{Q}^* \middle| \begin{matrix} \mathbf{g}_{q_1} \frown \ldots \frown \mathbf{g}_{q_K} \\ \boldsymbol{\varphi}_{q_1} \frown \ldots \frown \boldsymbol{\varphi}_{q_K} \end{matrix} \right\}. \tag{8.2}$$

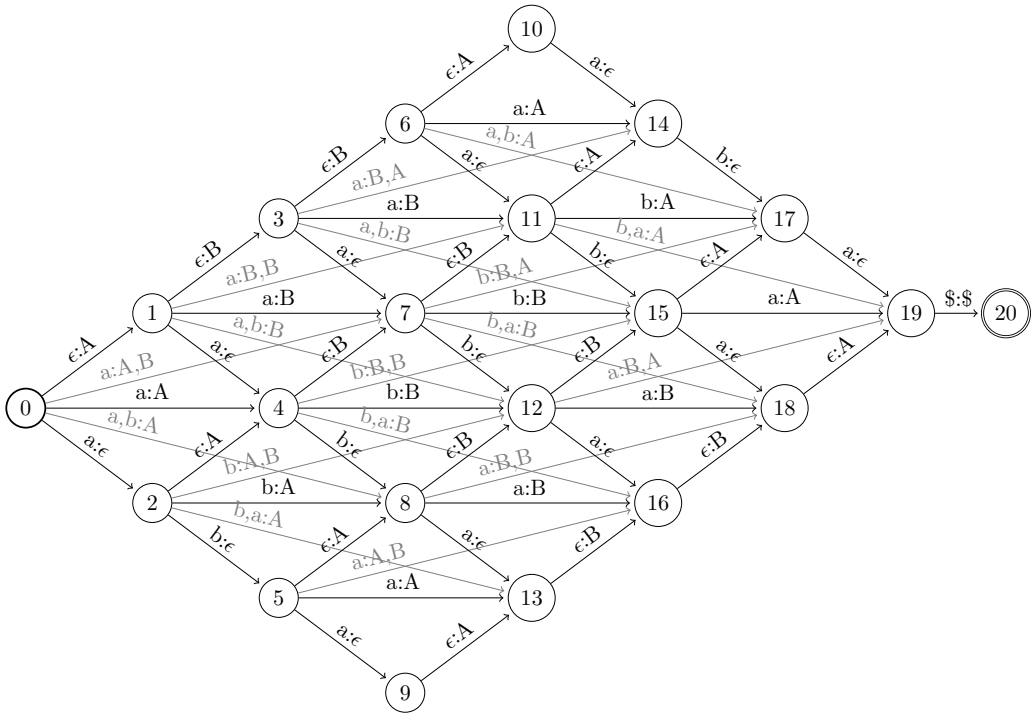


Figure 8.2: Lattice of all possible co-segmentations of letters $\mathbf{g} = A, B, B, A$ and phonemes $\boldsymbol{\varphi} = a, b, a$. Each vertex corresponds to a pair of positions in $\mathbf{g}$ and $\boldsymbol{\varphi}$, possibly conditioned on the history of graphones h. Edges correspond to graphones. Black: 01-to-01 graphones; Gray: additional 0..2-to-0..2 graphones (except $(2, 2)$).

The set of all possible alignments S can be represented as a lattice (Fig. 8.2). The joint probability $p(\mathbf{g}, \boldsymbol{\varphi})$ is determined by summing over all matching graphone sequences:

$$p(\mathbf{g}, \boldsymbol{\varphi}) = \sum_{\mathbf{q} \in S(\mathbf{g}, \boldsymbol{\varphi})} p(\mathbf{q}). \tag{8.3}$$

$p(\mathbf{g}, \boldsymbol{\varphi})$ is thus a probability distribution $p(\mathbf{q})$ over graphone sequences $\mathbf{q}_1^K = q_1, \ldots, q_K$, which can be modeled using a graphone language model (LM), using the standard M-gram approximation:

$$p(\mathbf{q}_1^K) \cong \prod_{j=1}^{K+1} p(q_j | q_{j-1}, \ldots, q_{j-M+1}). \tag{8.4}$$

To obtain a Bayesian joint-sequence model, we have to replace interpolated Kneser-Ney used in [62] with a Bayesian LM. In section 8.5, we introduce the hierarchical Pitman-Yor Process LM for that purpose. The task of G2P is to search for the most likely pronunciation given the orthographic form using Bayes' decision rule:

$$\boldsymbol{\varphi}(\mathbf{g}) = \arg \max_{\boldsymbol{\varphi}' \in \Phi^*} p(\mathbf{g}, \boldsymbol{\varphi}') \tag{8.5}$$

8.4 Model Estimation: Discounted EM

Many G2P algorithms require the grapheme-phoneme alignment (segmentation) as external input. Joint-sequence models have the advantage, that the alignment and the model parameters are optimized jointly on the training data $\mathcal{O} = (\mathbf{g}_1, \boldsymbol{\varphi}_1) \ldots (\mathbf{g}_N, \boldsymbol{\varphi}_N)$. The parameters to be estimated are the graphone M-grams $p(q_j|h_j;\boldsymbol{\vartheta})$ in Eq. 8.4, where $h_j = q_{j-1}, \ldots, q_{j-M+1}$ and $\boldsymbol{\vartheta}$ indicates a particular setting of the parameters. The training is performed with an Expectation-Maximization algorithm (EM) [62]:

$$e(q,h;\boldsymbol{\vartheta}) = \sum_{i=1}^{N} \sum_{\mathbf{q} \in S(\mathbf{g_i},\boldsymbol{\varphi}_i)} \frac{p(\mathbf{q};\boldsymbol{\vartheta})}{\sum_{\mathbf{q}' \in S(\mathbf{g_i},\boldsymbol{\varphi}_i)} p(\mathbf{q}';\boldsymbol{\vartheta})} \, n_{q,h}(\mathbf{q}) \tag{8.6}$$

In this expectation step, $e(q,h;\boldsymbol{\vartheta})$ is the expected number of occurrences (fractional count) of the graphone q in context h given the current parameters $\boldsymbol{\vartheta}$, and $n_{q,h}(\mathbf{q})$ is the number of times the particular graphone q occurs in the sequence $\mathbf{q}$. If we represent the set of all possible alignments as a lattice (Fig. 8.2), where the arc costs correspond to $p(q_j|h_j;\boldsymbol{\vartheta})$, Eq. 8.6 is summing the arc posteriors of the alignment lattice, which can be obtained with the lattice forward-backward algorithm. In maximum likelihood training, we start with a flat initialization of all possible graphones and we alternate the expectation and the maximization steps (Eqs. 8.6 and 8.7):

$$p(q|h;\boldsymbol{\vartheta'}) = \frac{e(q,h;\boldsymbol{\vartheta})}{\sum_{q'} e(q',h;\boldsymbol{\vartheta})} \tag{8.7}$$

The use of this original EM guarantees that the likelihood on the training set reaches a (local) optimimum, but it has several problems: it over-fits the training data, results in a huge graphone inventory, and once in any iteration $e(q,h;\boldsymbol{\vartheta}) = 0$, the graphone $q|h$ can never emerge again in future iterations. To avoid over-fitting and to keep the set of graphones manageable, the evidence counts are smoothed and pruned. As explained in [62], smoothing in this case needs to deal with fractional counts and an interpolated Kneser-Ney (KN) LM is used:

$$p_M(q|h) = \frac{\max\left(e(q,h) - d_M, 0\right)}{\sum_{q'} e(q',h)} + \lambda(h) \cdot p_{M-1}(q|\bar{h}) \tag{8.8}$$

Here, d_M is the discount used for model order M, $\lambda(h)$ is the interpolation weight, and $p_{M-1}(q|\bar{h})$ is the lower-order distribution (using a shortened history $\bar{h}$), which recursively has exactly the same shape as p_M, but uses a different kind of evidence counts $\hat{e}(q,\bar{h})$ according to a marginal constraint (details in [62]).

The discounted EM algorithm as implemented in Sequitur [62] is:

1. Initialize all graphones (flat).
2. Compute expected counts (Eq. 8.6).
3. Estimate new parameters (Eq. 8.8).
4. If likelihood on held-out set improved, continue with 2.
5. Tune discounting parameters $d_1, \ldots, d_M$ by optimizing the held-out likelihood.
6. If held-out likelihood improves, continue with 2.
7. Prune model and terminate.

While in the original EM (Eqs. 8.6 and 8.7), the training likelihood is guaranteed to converge to a (local) optimimum, for discounted EM, there are effectively two possibly conflicting objective functions: the setting of the optimal discount parameters (estimated on the held-out set) can in some cases detoriate the training likelihood and prevent the discounted EM from converging, causing a

sub-optimal termination of the overall algorithm. In order to reach the (local) optimum, it is in some cases necessary to manually keep the discounts small in the first few iterations until the training data 'guides' the model towards the optimum, and to apply the discounts only in the fine-tuning phase. As already pointed out in [62], starting from a larger initial graphone set (e.g. 0..2-to-0..2 graphones) always gave worse performance than when allowing only 01-to-01 graphones. Since those are a subset of the larger set, the training algorithm should be able to pick at least the same optimum.

Sequitur uses a bottom-up model construction: Starting with unigrams, the lower-order $M-1$ model is trained until convergence, and then the higher-order model $p_M(q|h)$ is initialized with $p_{M-1}(q|\bar{h})$ (called 'ramping-up'). Here, histories h can only be constructed from $\bar{h}$ that were not pruned in the lower-order model. This greedy approximation is necessary to keep the model tractable for higher orders M and when using graphones with more than one letters and phonemes. Surprisingly, when starting directly with a higher-order model (e.g. bigram), the training finishes in a worse local optimum, than when training bottom-up (fixing the optimal set of graphones in the unigram, and training a bigram on top of that). That indicates, that the training procedure is not able to find good sets of unigram graphones, even if the bigger context should help to make an even better selection.

As seen in this section, the implementation of the discounted EM needs a good deal of engineering, and sometimes it is necessary to force the model into the right direction. We therefore propose to replace the smoothed graphone LM with a Bayesian LM, and to train the model in more principled, fully Bayesian way.

8.5 Hierarchical Pitman-Yor Process LM

To obtain a Bayesian joint-sequence model, we use a non-parametric Bayesian LM as graphone LM for the computation of $p(q|h;\boldsymbol{\vartheta})$ instead of Eq. 8.8. We chose the Hierarchical Pitman-Yor Process language model (HPYLM) [75, 76, 77], that has achieved good results as word LM and results in a similar form as the interpolated KN (Eq. 8.8), which can be interpreted as a HPYLM with a special form of inference [76]. The hidden variables in a HPYLM are the distributions of graphones given a particular context $p(q|h;\boldsymbol{\vartheta})$, and they are related to each other in a hierarchical structure, where the prior mean (base measure) of a particular context is given by the distribution of graphones in the shortened context ($\bar{h}$, leaving out the earliest graphone). This hierarchical structure (Fig. 8.3) corresponds exactly to interpolating between higher and lower order n-grams.

Each hidden variable $p(q|h;\boldsymbol{\vartheta})$ is distributed according to a Pitman-Yor process (PY), which is a generalization of the Dirichlet process. A PY generates a probability distribution G (in our case discrete, over graphones), that is similar to another distribution G_0 called base measure. The distribution $G \sim PY(d,\theta,G_0)$ has two parameters: the discount factor d, which shapes the tail of the distribution and θ controling the similarity of G to G_0. At the lowest hierarchy level, the base measure for unigrams $G_0 = 1/|\mathcal{Q}|$ is a uniform distribution over graphones:

$$\begin{aligned} G_1 &\sim PY(d_1,\theta_1,G_0 = 1/|\mathcal{Q}|) \\ &\dots \\ G_{\bar{h}} &\sim PY(d_{|\bar{h}|},\theta_{|\bar{h}|},G_{\bar{h}}) \\ G_h &\sim PY(d_{|h|},\theta_{|h|},G_{\bar{h}}) \end{aligned}$$

It is not possible to observe G, since it has an infinite number of components. However, there is an equal representation of the HPYLM with G integrated out, in the form of a hierarchy of Chinese restaurant processes (Fig. 8.3). There is one Chinese restaurant process for each graphone q in context h (including the empty context $\emptyset$ for unigrams). The training of the model is done by seating customers (graphone n-gram counts $c(q|h)$) over tables $1 \dots t_{hq}$ (the number of tables for a particular graphone/context). We use Gibbs sampling, where one sampling step is to remove a customer and to

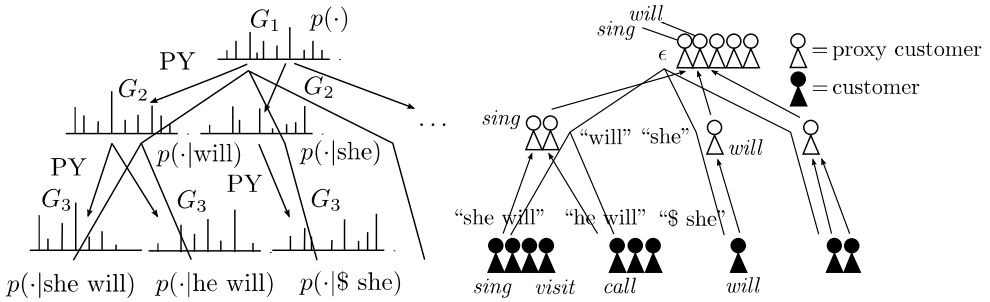


Figure 8.3: Hierarchical Pitman-Yor Process language model and its corresponding hierarchical Chinese restaurant processes [78].

re-seat the customer by choosing a table k:

$$k = \begin{cases} c_{hqk} - d & (k = 1 \dots t_{hq}) \\ \theta + d \cdot t_{h\cdot} & (k = new). \end{cases}$$

Here, c_{hqk} is the number of customers at table k so far, and $t_{h\cdot} = \sum_q t_{hq}$. The customers $c(q|h)$ are only directly seated at the highest order M. Every time a new table k is created, a proxy-customer is hierarchically sent down (Fig. 8.3). Therefore, the lower-order distributions are only updated for graphones in unseen contexts, and as in KN smoothing, they are not proportional to counts $c(q|\bar{h})$. The resulting equation for the graphone HYPLM (with $\theta = 0$ and $t_{hq} = 1$) resembles the interpolated KN (Eq. 8.8):

$$p(q|h) = \frac{c(q|h) - d \cdot t_{hq}}{\theta + c(h)} + \frac{\theta + d \cdot t_{h\cdot}}{\theta + c(h)} \cdot p(q|\bar{h}) \tag{8.9}$$

There are two hidden variables in the inference of the graphone HPYLM: the co-segmentation $S(\mathbf{g}, \boldsymbol{\varphi})$ of the grapheme/phoneme sequence and the seating arrangements of the Chinese restaurants. A direct implementation of Gibbs sampling would sample one boundary of a single graphone at a time (as done in [68]), which results in an inefficient algorithm that can only take into account local (bigram) statistics. Instead, we use a blocked Gibbs sampler, i.e. we sample the co-segmentation $S(\mathbf{g}, \boldsymbol{\varphi})$ of a whole utterance (word), re-seating all corresponding customers at once. Sampling means to select one path (graphone sequence) in the alignment lattice (Fig. 8.2) according to the posterior probability $p(\mathbf{q}|\mathbf{g}, \boldsymbol{\varphi}; \boldsymbol{\vartheta})$. As shown in [68] and [78] this can be implemented with the forward filtering and backward sampling procedure. Forward filtering is the forward part of the lattice forward-backward algorithm (as in Section 8.4) and backward sampling picks a path according to the forward probabilities, starting from the final state. Thus, the inference procedure in the graphone HPYLM is to iterate over all training utterances (words):

1. Sample co-segmentation $S(\mathbf{g}, \boldsymbol{\varphi})$ according to posterior.
2. Update graphone counts $c(q, h)$.
3. Sample seating arrangements t_{hq} in Chinese rest. (Fig. 8.3).

8.6 Implementation with WFSTs

We implemented the whole Bayesian G2P framework with the help of weighted finite state transducers (WFST) [79] mostly using standard library functions of OpenFST `www.openfst.org/`. The generation of all possible graphone segmentations in an alignment lattice can be implemented using WFST composition.

As shown in Fig. 8.4, the letter sequence $\mathbf{g}$ and the phoneme sequence $\boldsymbol{\varphi}$ can be represented as linear acceptors L and P, respectively. To construct a lattice containing all possible alignments, we

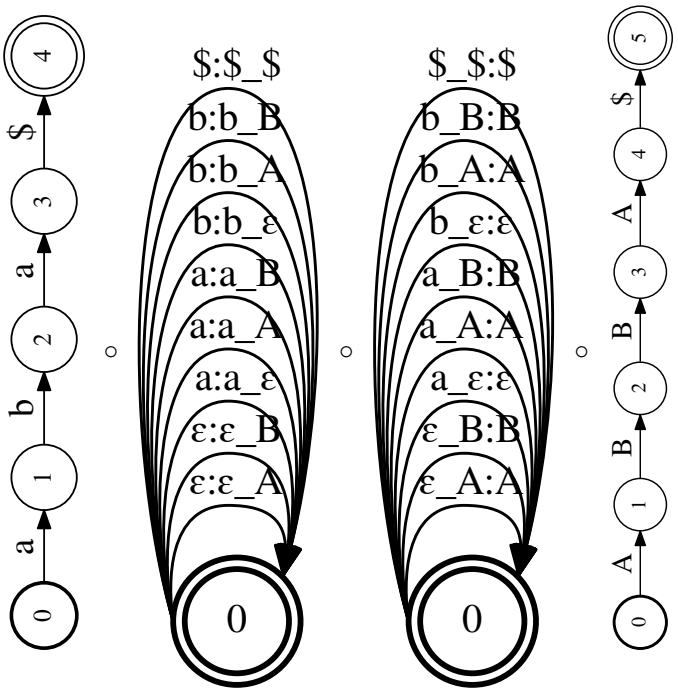


Figure 8.4: Transducer chain $P \circ P2G \circ G2L \circ L$ for toy example with grapheme inventory A, B and phoneme inventory a, b. Outer left: phoneme acceptor P for $\boldsymbol{\varphi} = a, b, a$; Outer right: letter acceptor L for $\mathbf{g} = A, B, B, A$ corresponding to a pronunciation dictionary entry '$ABBA\ a\,b\,a$'. Middle part: Left: transducer $P2G$ mapping from phonemes to the set of all possible graphones. Right: $G2L$ mapping from graphones to letters.

use two mapping transducers. In Fig. 8.4, transducer $P2G$ (middle left) maps from graphones to phonemes and transducer $G2L$ (middle right) maps from graphones to letters. For simplicity, we use only 01-to-01 graphones (Fig. 8.1), so the set of all possible graphones stays reasonable. Given these transducers, we can form a chain of compositions to produce the alignment lattice transducer (example in Fig. 8.4 results in Fig. 8.2.): $A = P \circ P2G \circ G2L \circ L$.

We use a blocked Gibbs sampling approach, where we always sample a new alignment for a whole pronunciation entry (word) at once. A sample alignment is a particular path through the lattice A (Fig. 8.2). Also the graphone LM (HPYLM) can be represented as a WFST G.To represent an n-gram LM as WFST, we use the compact representation using back-off arcs ([79], page 19). We can apply the probabilities of the graphone HPYLM with the help of WFST composition (which corresponds to lattice re-scoring):

$$B = P \circ P2G \circ G \circ G2L \circ L \tag{8.10}$$

As already pointed out by [71], to correctly evaluate the interpolated LM in the WFST framework, we need to encode the back-offs as failure arcs [80] and to use the correct matchers in the composition (phi-composition, indicated by $\circ_\varphi$). For higher-order graphone LMs, and already for small graphone inventories, the G transducer gets huge. Moreover, for a particular training utterance (word), only a small portion of G is accessed. Therefore, we use OpenFST's interface for lazy composition. We implemented the HPYLM with the source code developed by Walter/Heymann [81] `https://github.com/fgnt/nhpylm` and wrote our own wrapper, that creates a lazy (on-the-fly) OpenFST WFST object.

While WFST composition is an associative operation, the grouping of compositions in Eq. 8.10 has an important impact on memory use and speed, especially when using lazy composition (and possibly pruning). Since the composition with G is the most costly operation and the linear acceptors P and L are the knowlegde sources that constrain the possible graphone sequences, we want to apply them

as early as possible, before applying G. The final composition is:

$$B = \Pi_2(\Pi_1(P \circ P2G) \circ G2L \circ L) \circ_\varphi G \tag{8.11}$$

The projection operations $\Pi_1(T)$ and $\Pi_2(T)$ obtain an acceptor from WFST T by omitting the input or output labels, respectively. We project onto the output symbols after composing $P \circ P2G$ and project the resulting alignment lattice A onto the input symbols to obtain an acceptor lattice with graphone labels. Eq. 8.11 results in a 2-3x speed-up over Eq. 8.10. In the WFST framework, the forward filtering and backward sampling procedure as used in [68] and [78] can be implemented by applying weight pushing towards the initial state in the (log) probability semi-ring, and then forward-sampling a path (graphone sequence), which is used to sample a new seating arrangement in the Chinese restaurant processes. During training, we go through all training utterances in random order. Typically, 3-4 iterations through the data are sufficient to converge to a likely segmentation and seating arrangement. After each iteration, we re-sample the hyperparameters for d and ϑ as described in [76], appendix C. Since we use Gibbs sampling to approximate $p(q|h)$, correct estimates can be obtained by averaging several HPYLM with different seating arrangements. As a first approximation, we used just a single HPYLM in the experiments.

8.7 Experimental results and conclusions

We trained the HPYLM G2P on the CMUdict v0.7 kindly provided by [71]. It contains 106,837 unique training words with 113,438 pronunciations. The test set contains 12,000 unique words with 12,753 pronunciations. Our baseline is a 7-gram joint-sequence model trained with Sequitur [62] using the default settings and selected 1% of the training as held-out set to tune the discounts. We reached 5.92% phoneme error rate (PER) and 24.65% word error rate (WER) after 11h of training, which is very close to what is reported in [62]. Using a 9-gram LM as in [62] took an additional 9h training and gave the same performance. Our Bayesian HPYLM G2P does not need a held-out set. After three iterations of sampling the training set in 2h, we reached 5.92% PER and 24.73% WER, which is basically the same as our baseline. We can expect further improvement from averaging several sampled HPYLM. With Phonetisaurus [71] we reached 5.80% PER and 24.36% WER in the order of minutes.

We presented a fully Bayesian approach to G2P, which is fully implemented with WFSTs. The Bayesian G2P based on a hierarchical Pitman-Yor-Process does not need a held-out set and complicated parameter tuning and avoids the pitfalls of the discounted EM algorithm. The Bayesian model has the same performance as the smoothed joint-sequence models. Despite the fact, that Gibbs sampling was used and the resulting models (7-grams) are already significantly large, the training is much faster than using Sequitur, but still slower than Phonetisaurus. No greedy assumptions are necessary, as e.g. the bottom-up model initialization ([62]) and the segmentation is done jointly in training, using full context. However, the most important advantage is that the resulting model can be used in a bigger Bayesian framework and can deal with (partly) un-annotated data.

Chapter 9

Unsupervised learning of pronunciation dictionary from unaligned phone and word data

Takahiro Shinozaki, Shinji Watanabe, Daichi Mochihashi and Graham Neubig

The performance of automatic speech recognition systems has recently approached human levels in several tasks. However, there is still a large gap in the handling of unknown words. Humans are able to recognize words, even ones they have never heard before, by reading text and understanding the context in which a word is used. While this ability is important to keep updating the vocabulary for new words that appear daily, existing methods based on G2P or OOV detection lack a holistic mechanism of learning the pronunciation and spelling of new words from textual and acoustic evidence. In this work, we propose a new paradigm in learning for speech recognition that parallels the human ability to learn new words by reading text: automatic learning of word pronunciations from unaligned acoustic and textual data. While the task is very challenging and we are at the initial stage, we demonstrate that a model based on Bayesian learning of Dirichlet processes can acquire word pronunciations from phone transcripts and text of the WSJ data set.

9.1 Introduction

While the recognition accuracy of automatic speech recognition systems is approaching the human level, the performance is heavily dependent on supervised learning. To support a new task domain or new words, which are invented daily, new labeled speech data and pronunciations of new words must be prepared. This often limits the usability of the system to the initially prepared domain due to the large cost.

Compared to the use of unlabeled speech data for acoustic model training [99, 100, 101], there are relatively few studies about automatic acquisition of word pronunciations. However, this learning of pronunciations is an essential step in creating adaptable speech recognition system that requires less human help.

An existing approach to find a mapping from a word to its pronunciations is *grapheme to phoneme (G2P) conversion* [102, 103, 104, 105], where a pre-trained G2P converter is applied to the surface form of a new word. A limitation is that these methods are not applicable for words for which the pronunciation is hard to infer directly from the spelling. While there are several works that learn pronunciations from acoustic data (e.g. [106]), these works generally assume parallel speech and text data, which is hard to come by.

Another approach is based on *out of vocabulary (OOV) detection* [107, 108] and phone recognition. Speech input is first decoded by a phone recognizer, and a speech segment detected as an OOV is labeled by the decoded phone sequence at the corresponding time position. By combining with a word decoder, it is expected that a word is output if it is included in the decoders' vocabulary and otherwise a phone sequence is output [109, 110]. This method has the flexibility to discover unknown words automatically, but is not able to take advantage of external textual resources, and has no way of connecting the phoneme string with its actual spelling in natural written language.

The source of information about new words is from the word-level text for the G2P approach, and from phoneme recognition results for the OOV detection based approach. In this chapter, we propose a new paradigm of learning from unaligned speech and text, which allows us to utilize information about new words from both of these sources simultaneously. This is done by creating a probabilistic model of the pronunciation dictionary and performing Bayesian inference [111] to estimate its parameters, under the assumption that the text and speech are from the same mother distribution, which corresponds to the language at hand. By making this assumption, we can use distributional information found in text to guess which pronunciation corresponds to which word. The more instances of a word with unknown pronunciation that appear in the phone transcript, the fewer the possible pronunciations of the word will become. As a result, the approach can learn words from unaligned speech and textual context, allowing it to improve over the OOV detection approach by giving a mapping from a word spelling to its pronunciations, and giving it advantages over pure G2P by allowing it to learn from acoustic evidence.

The organization of the chapter is as follows. Firstly the proposed method is explained in section 9.2. Then experimental condition is described in Section 9.3 and the results are shown in Section 9.4. Finally, conclusion and future works are given in Section 9.5.

9.2 Proposed unsupervised pronunciation dictionary learning method

The proposed method is based on treating pronunciation dictionary as a random variable. In the followings, the formulation of the pronunciation dictionary is first explained. Then a Bayesian network representation of the overall framework of the proposed method is given. Finally, a weighted finite state transducer (WFST) [112] based implementation is introduced.

9.2.1 Probability model of a pronunciation dictionary

The proposed pronunciation dictionary model is a conditional probability $\mathrm{P}\left(p|w\right)$ of a pronunciation p (e.g. "HH AH L OW") given a word w (e.g. "hello"). The pronunciation is a finite length phone sequence. The pronunciation dictionary is represented by an array of words each of which is associated with an infinite distribution of pronunciations to potentially allow any pronunciation for a word. Figure 9.1 depicts the structure of the pronunciation dictionary. The distribution of the pronunciation of a word is generated by a Dirichlet process [113]. The probability of the pronunciation dictionary is the joint probability of the pronunciation distributions as shown in Equation (9.1).

$$\mathrm{P}\left(PD\right) = \prod_{w \in V} DP\left[\mathrm{P}_w\left(p\right)|\alpha, G_0\right], \tag{9.1}$$

where PD is the pronunciation dictionary, V is the vocabulary, $DP\left[\mathrm{P}_w|\alpha, G_0\right]$ is the probability that a distribution of a pronunciation $\mathrm{P}_w\left(p\right)$ is generated from a Dirichlet process with a concentration parameter α and a base distribution G_0. A draw from the base distribution G_0 is a pronunciation. All the Dirichlet processes share the same base distribution.

While each word has infinite number of pronunciations in our modeling, the trick of the Dirichlet process is that the predictive distribution given a finite amount of observations is explicitly obtained

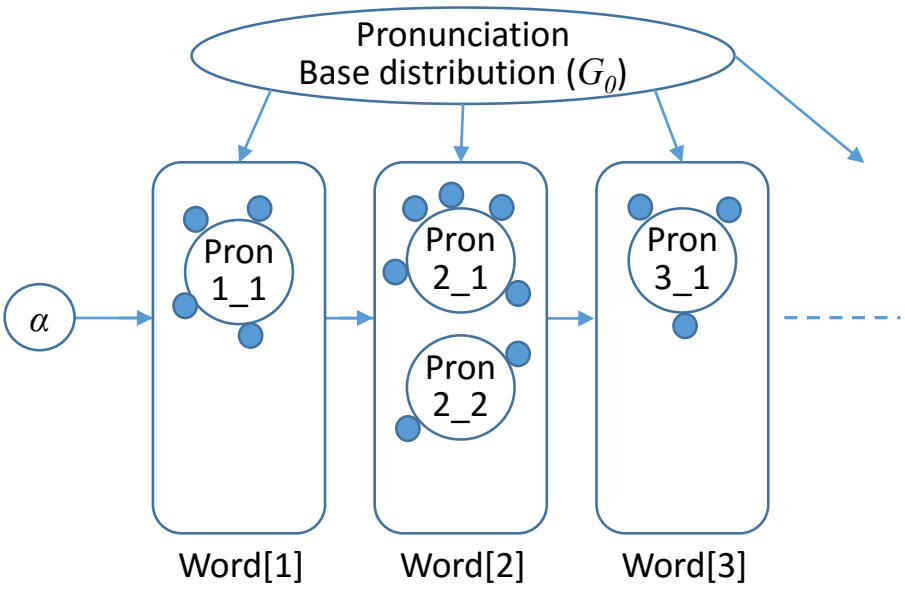


Figure 9.1: Proposed pronunciation dictionary model. A word entry is represented by a no corner rectangle and it corresponds to a restaurant of the Chinese restaurant process. An open circle in the restaurant is a table that represents a pronunciation, and a filled small circle is a customer that corresponds to an appearance of the pronunciation.

by the Chinese restaurant process [114]. For our pronunciation dictionary, a word in the dictionary is a restaurant, and an appearance of a pronunciation in data is an appearance of a customer. Let's assume that a set of utterances U are observed in which a word w has appeared n_w times. Let's also assume that a pronunciation p of the word w has appeared n_p times where $\sum n_p = n_w$, and the number of appearances of different pronunciations for w is m_w. Then, the predictive distribution of the pronunciation p for the word w is given by Equation (9.2).

$$\mathrm{P}\left(p|w,U\right) = \frac{n_p}{\alpha + n_w} + \frac{\alpha}{\alpha + n_w} G_0\left(p\right). \tag{9.2}$$

Since the number of non-zero observation counts n_p is m_w, and m_w is at max n_w, only limited amount of memory is needed to store the value of n_p. The fact that a word usually has only a few (mostly only one) pronunciations is represented by choosing α close to 0.0.

9.2.2 Bayesian network based system modeling for training and evaluation

If aligned phone transcripts with word boundaries and word level text are given, then pairs of a word and its pronunciation are easily extracted and the estimation of the pronunciation dictionary is trivial. However, if they are not aligned and the word boundary is unknown in the phone transcript, the problem becomes much more difficult.

To perform unsupervised learning and evaluation of the pronunciation dictionary using unaligned phone transcripts and word level text, we use Gibbs sampling [115, 116] for a full Bayesian approach. The probabilistic inference is performed based on a Bayesian network shown in Figure 9.2. The definitions of the nodes in the network are summarized in Table 9.1. The shown network is a large view, and each node has internal structures.

As is described in the table, the node "Language model" is a hierarchical Bayesian language model [117]. The node "Pronunciation dictionary" represents the proposed pronunciation dictionary. The node represented by a filled small circle represents a rule to convert the word segmented phone sequence to no word segmented phone sequence, which correspond to the hidden semi-Markov model [118]. It is a constant and not a random variable.

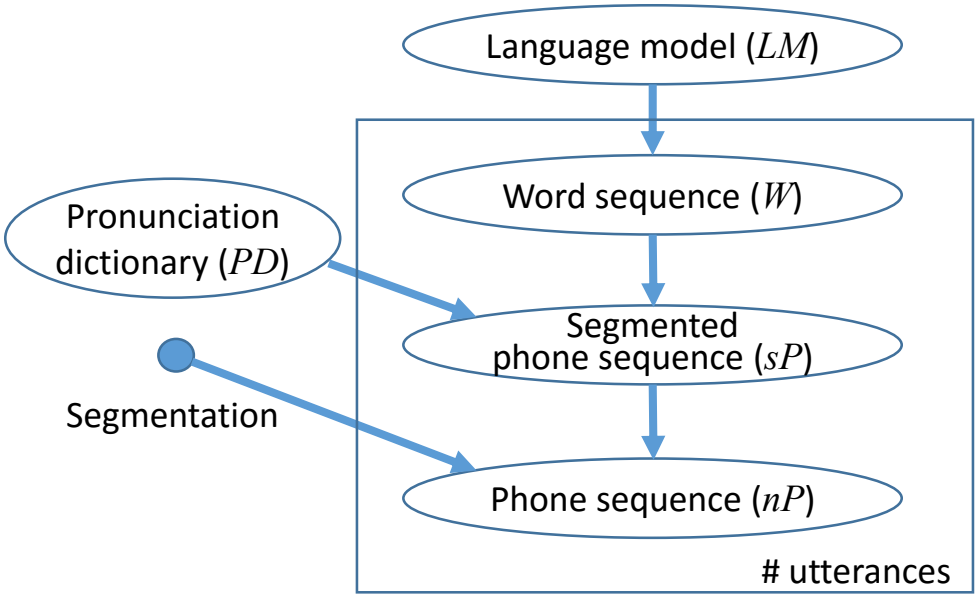


Figure 9.2: Bayesian network representing the full system to train and evaluate the pronunciation dictionary using unaligned phone and word data.

Table 9.1: Description of the nodes in the Bayesian network shown in Figure 9.2.

Node	Description
Pronunciation dictionary (PD)	Pronunciation dictionary based on a set of Dirichlet distributions
Language model (LM)	Hierarchical Bayesian language model
Word sequence (W)	Word sequence of an utterance
Segmented phone sequence (sP)	Phone sequence of an utterance with word boundaries
Phone sequence (nP)	Phone sequence of an utterance without a word boundary

The three nodes "Word sequence", "Segmented phone sequence", and "Phone sequence" represent an utterance given as a word sequence, a word segmented phone sequence, and a phone sequence with no word segmentation, respectively. The word sequence is generated from the language model, and the segmented phone sequence is generated from the word sequence and the pronunciation dictionary by replacing the word entries to their pronunciation based on the distributions of the pronunciations. The no word segmented phone sequence is generated from the word segmented phone sequence. For example, "the sale of the hotels" is a word sequence, "DH AH </w> S EY L </w> AH V </w> DH AH </w> HH OW T EH L Z </w>" is a word segmented phone sequence where </w> represents a word boundary, and "DH AH S EY L AH V DH AH HH OW T EH L Z" is a no word segmented phone sequence. Each utterance is assumed to be independent given the language model and the pronunciation dictionary.

The framework is an extension of the unsupervised word segmentation. If the phones and characters are the same and the pronunciation dictionary is replaced with a spelling model, then it reduces to the unsupervised word segmentation [119, 120].

9.2.3 Gibbs sampling for learning and evaluation

The Bayesian network shown in Figure 9.2 has three nodes to represent an utterance. When an utterance is given as a word level text, the word sequence node of the utterance is observed and the word segmented phone sequence and the phone sequence nodes are hidden. Similarly, when an utterance is given as a phone level transcript, the phone sequence node is observed and the other two nodes are hidden. The Gibbs sampling is performed by repeating randomly picking up an utterance

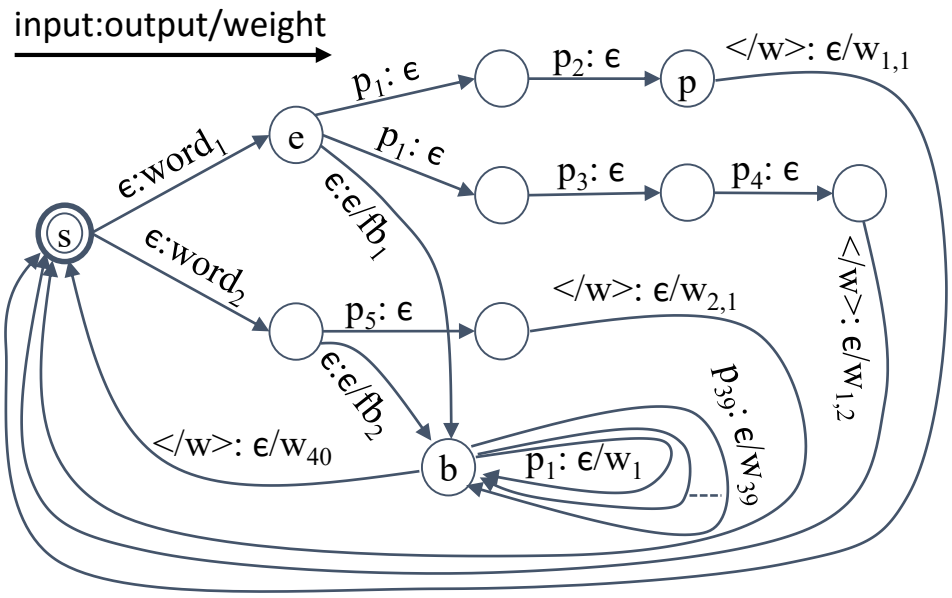


Figure 9.3: Example WFST of a pronunciation dictionary. Arc labels word_l and p_j represent word and phone symbols, respectively.

and updating the values of its hidden nodes by drawing a sample from their posteriors given values of the rest of the nodes starting with an initial assignment.

The joint posterior of the hidden nodes of a selected utterance is obtained from a joint posterior of the three nodes, which is obtained by Equation (9.3).

$$\begin{aligned} & \mathrm{P}\left(nP_s, sP_s, W_s | nP_T, sP_T, W_T\right) \\ = & \int \mathrm{P}\left(nP_s, sP_s, W_s, LM, PD | nP_T, sP_T, W_T\right) dLMdPD \\ = & \mathrm{P}\left(W_s | W_T\right) \mathrm{P}\left(sP_s | W_s, W_T, sP_T\right) \mathrm{P}\left(nP_s | sP_s\right), \end{aligned} \tag{9.3}$$

where nP, sP, W, LM, PD represents the phone sequence, word segmented phone sequence, word sequence, language model, and pronunciation dictionary, respectively. The suffix s indicates the selected utterance, and T indicates the set of rest utterances. The derivation of Equation (9.3) is based on the Bayes chain rule, conditional independences that are read from the Bayesian network by d-separation [121], and marginalization of LM and PD which are obtained based on the Chinese restaurant process. The posterior of the word segmented phone sequence of the selected utterance $\mathrm{P}\left(sP_s | W_s, W_T, sP_T\right)$ can be easily evaluated by the Chinese restaurant processes because both W_T and sP_T are in the conditional part, which means they are treated as if they were observed at the same time for the same utterances in T, which in tern means they are treated as if their alignments were known. When an utterance is given as the word level text, the segmented phone sequence node and the phone sequence node may be marginalized out instead of sampling their values because of the Bayesian network structure.

9.2.4 WFST based implementation

The sampling from the joint posterior distribution of the hidden variables of the selected utterance that is obtained from Equation (9.3) is not a simple task since the nodes have complex internal structures similar to speech recognition. To implement the Gibbs sampling, we make use of the framework of the WFST, extending the implementation of the unsupervised word and LM learning [122, 123, 124] introducing the pronunciation dictionary.

For the sampling from $\mathrm{P}\left(sP_s, W_s | nP_s, nP_T, sP_T, W_T\right)$, first the input phone sequence of the selected utterance nP_s and each component of Equation (9.3) are represented by WFSTs and they are composed

to form a single WFST that expresses an unnormalized distribution of the posterior. Then a sample is obtained by applying the forward filtering backward sampling algorithm to the composed WFST. The WFST for $\mathrm{P}\left(nP_s|sP_s\right)$ takes a phone sequence nP_s as the input and outputs a word segmented phone sequence sP_s. All possible segmentations are encoded as WFST paths. The construction of the WFST for $\mathrm{P}\left(W_s|W_T\right)$ is basically the same as for the N-gram WFST. It takes a word sequence W_s as the input and outputs the same word sequence assigning a probability. For the details about these two WFSTs and the sampling, please refer to [123]. Here, only the differences are described.

The WFST of $\mathrm{P}\left(sP_s|W_s, W_T, sP_T\right)$ corresponds to a pronunciation dictionary, and transduces a word segmented phone sequence sP_s to a word sequence W_s. It has a structure consisting from a normal pronunciation dictionary and a sub module encoding the base distribution. Figure 9.3 shows an example of the pronunciation dictionary WFST having only two words. For each word in the vocabulary, an arc starts from the start node (s) having the word as the output label. Each pronunciation of the word associated with a table of the Chinese restaurant process is represented by a path having the pronunciation as a sequence of phone input labels, where the path starts from the end node of the word arc (e). The end of the pronunciation path (p) is connected to the start node by an arc with a word boundary input symbol and an weight $w_{w,p} = \frac{n_p}{\alpha+n_w}$ that corresponds to the first term of Equation (9.2), where n_w and n_p are the counts of the word and the pronunciation obtained from W_T and sP_T. There is another arc with an weight $fb_w = \frac{\alpha}{\alpha+n_w}$ that starts from the end node (e) of the word arc and ends in a node (b), which corresponds to falling back to the base distribution G_0. In the figure, phone 1-gram is assumed as the base distribution and it is implemented by phone loops. Once a pronunciation is drawn from the base distribution, it goes back to the start node (s) by an arc having a word boundary symbol as the input. In real implementation, a more compact WFST can be made by adopting the tree structured dictionary.

When composing a WFST, a problem is that the intermediate symbols are removed. This means the necessary information about the segmented phone sequence sP_s is marginalized out when composing the WFSTs of the input phone sequence, the segmentation, the pronunciation dictionary and the language model. To address the problem and obtain a value for sP_s as the result of the sampling, we modify the composition operation so that intermediate symbols are accumulated in the input label. When an arc having "a" and "b" as the input and output labels and an arc having "b" and "c" are composed by the modified composition, the composed arc has "a_b" as the input label and "c" as the output label instead of "a" and "c".

9.3 Experimental setup

Experiments were performed using the WSJ corpus [125, 126] and the CMU dictionary. As the phone transcript, true phone labels were used. While the phone names adopted by the CMU dictionary are represented by a string, we truncated them to a single character removing all but the first character, which increased the ambiguity in the mapping from a phone sequence to a word sequence. The reduced phone set size was 25. The word entries of the pronunciation dictionary was made from a word level transcript, in which 85% were given a pronunciation as an initial setting. No pronunciation was assigned to the remaining 15% of the words. The task was to find their pronunciations from unaligned word and phone level transcripts. As the base distribution for the pronunciation, phone 0-gram was used. The length of the pronunciations were constrained to be greater than 0. The concentration parameter α for the pronunciation dictionary was set to 0.001. During the sampling using the phone level transcript, the vocabulary was fixed so that no new word was generated with unknown spelling. The software was implemented by modifying the LatticeWordSegmentation [124, 127, 122].

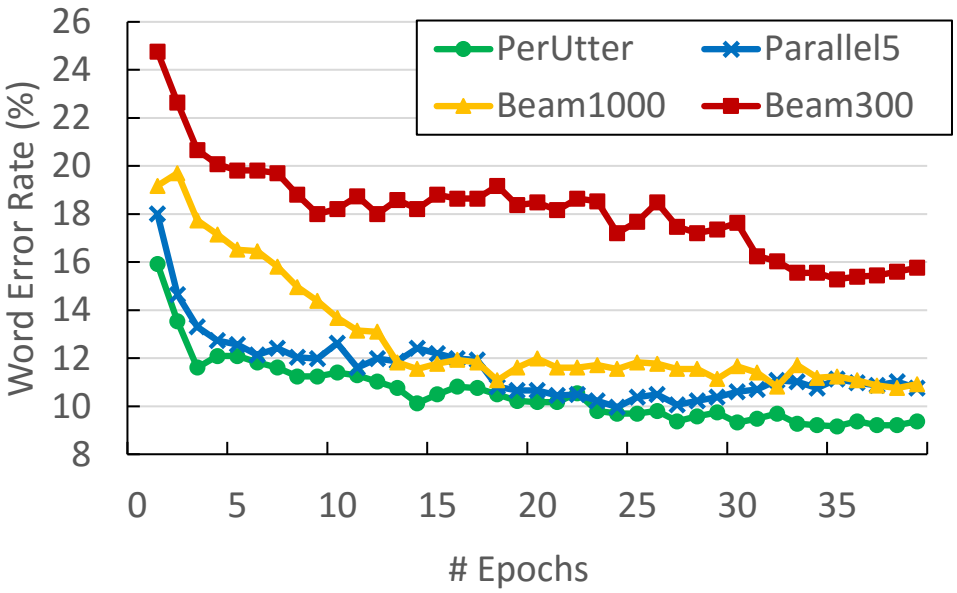


Figure 9.4: Word error rate when using 2-gram language model.

9.4 Results

Figure 9.4 shows the relation of the number of epochs of the Gibbs sampling and the word error rate (WER) when 100 utterances with phone level transcript and 100 utterances with word level text were used. The word and the phone level transcripts were obtained from the same 100 utterances in the corpus. However, no utterance level alignment information was given to the system. The vocabulary size was 849 in which pronunciations were given to 742 words. The language model was a word 2-gram and the perplexity was 30.8. The WER was evaluated for output word labels of WFST paths obtained by the sampling for the phone input data. In the figure, "PerUtter" indicates that the sampling was performed utterance by utterance with a single thread processing. "Parallel5" means five utterances were processed in parallel and the variable update was performed gathering their statistics. "Beam1000" and "Beam300" were the results when the sampling was performed for a list of hypotheses obtained by a beam search for the purpose of reducing the memory and CPU requirements by combining the lazy WFST composition [112].

As can be seen, PerUtter gave the best result achieving the lowest WER of 9.2%. Parallel5 gave slightly worse but comparable result as PerUtter. When the beam search was used with the beam width 1000, the WER was larger than Parallel5 at the beginning but it became close after enough epochs. When the beam width of 300 was used, WER largely increased compared to the others. Although, it also gave some improvement compared to the beginning by repeating the epochs. When a Xeon E5-2630v2 CPU was used, wall-clock time of running PerUtter, Beam1000, and Beam300 were 161, 90, and 32 minutes per an epoch, respectively. When a Core i7 6800k CPU was used, it was 122 minutes for PerUtter and 42 minutes for Parallel5. Table 9.2 shows an example of a part of sampled word level text obtained for a phone transcript input when PerUtter was used. It can be seen the correct sentence was obtained at third epoch, which was the result of successful pronunciation assignment.

Figure 9.5 shows dictionary error rate, which is defined as a ratio of correct pronunciations in the sampled pronunciation dictionary. Insertion error means an extra wrong pronunciation is introduced, and deletion means correct pronunciation is missing. The total error rate is their sum. In this experiment, 3-gram language model was used and five utterances were processed in parallel. As can be seen, correct pronunciations were acquired by iterating the epochs. After 30 iterations, the minimum total error rate was 4.0%.

Table 9.2: Example of a part of sampled sentences. Pronunciations of the words “fed” and “assets” were unknown.

reference	strategy to sell off assets and
epoch1	strategy to sell a fuss eight fed and
epoch2	strategy to sell officer bids and
epoch3	strategy to sell off assets and

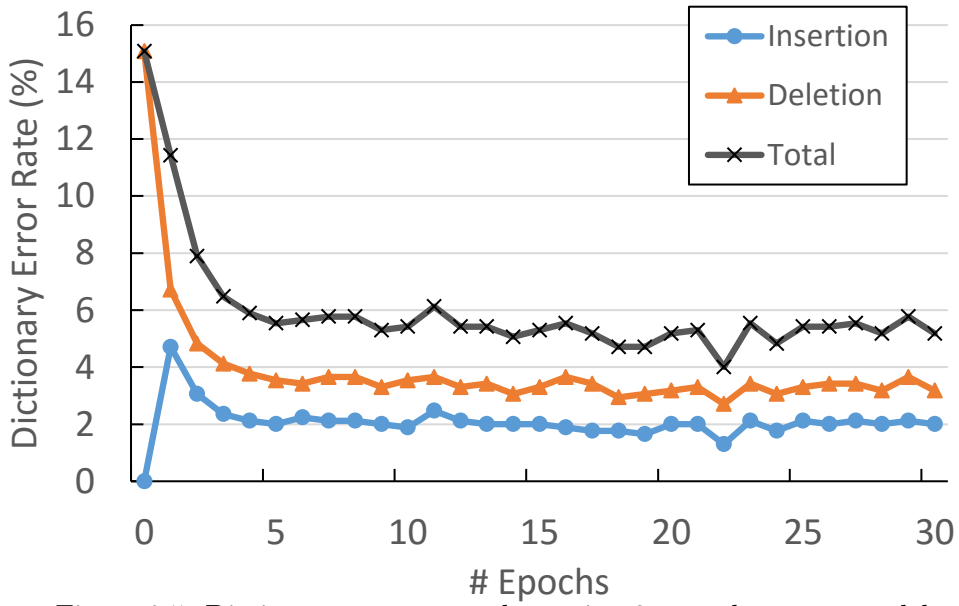


Figure 9.5: Dictionary error rate when using 3-gram language model.

Finally, an experiment was performed by using a phone and word level transcripts that were derived from non-overlapping sets of utterances in WSJ. To obtain reasonable word perplexity, the amount of word level transcription was increased to 10000 utterances. The vocabulary size was 13240 in which 11254 were given pronunciations as the initialization. Word 2-gram is used for the language model and the perplexity was 181. Due to the largely increased vocabulary and the LM sizes, running an experiment with this setting was quite tough requiring huge memory and CPU time. Therefore, we needed to combine the parallel sampling with the parallel factor of three and the beam search with the beam width 300. Because of the increased perplexity and the errors in the sampling, only a small improvement of WER from 29.3 to 28.0 was obtained. This result indicates the probability model and the sampling algorithm need to be improved to get good performance when the perplexity and the vocabulary size are large.

9.5 Conclusion and future works

We have proposed a new framework of unsupervised pronunciation dictionary learning using unaligned phone and word data. Experimental results show that it works well when perplexity and vocabulary size of the task is small, while it needs improvements in the modeling and the sampling algorithm to deal with more difficult tasks. Future works include combining with G2P to pick the best of both, and implementing the slice sampling [128] to improve the efficiency of the sampling.

Chapter 10

Graphemic Knowledge Transfer

Matthew Wiesner

This work describes two methods of cross-lingual knowledge transfer by exploiting the shared orthographies of high and low resource languages. In this sense the focus is more on transfer learning for corpora with no transcribed audio than on unsupervised techniques. We propose a simple framework for automatic speech recognition in a setting with no transcribed audio. The assumption of these methods is that most low-resource languages are written using a phonemic orthography shared by a high-resource language and that furthermore, shared graphemes in these orthographies have similar acoustic realizations.

10.1 Introduction

Prior work on cross-lingual knowledge transfer has generally focused on learning language independent acoustic features [27, 130, 131], normally through multilingual bottleneck features, or it has focused on exploiting shared linguistic knowledge encoded in the form of a universal phoneme set [129, 98]. These techniques have been used for language adaptation, or bootstrapping speech recognition systems in new languages by transferring acoustic models across languages [132].

One promising recent line of work [135] adapts universal phonemic acoustic models to a low-resource target language. Universal phonemic acoustic models are trained on many languages and then adapted by including noisy phonetic transcriptions of the target language. The phonetic transcriptions are the output of a WFST trained on transcriptions of phonetic sequences in the native orthographies of Turkers. In this sense, the WFST is just a G2P component used to generate phonetic transcriptions.

This work investigates a few architectures based roughly on the work in [135]. First, we replace the universal phoneme set with generic acoustic units learned in an unsupervised fashion on the target language. It is hoped that these acoustic units more accurately model the target language acoustics than the universal phoneme set. We have seen that using unsupervised acoustic units in the target language seems to perform better for topic-id and in terms of normalized mutual information with ground truth phonemes than using mismatched phoneme recognizers ported from other languages.

We also eliminate the role of Turkers primarily through the assumption of shared graphemes across languages. Prior work on graphemic speech recognition systems [134] shows that using graphemic acoustic models results in only minor degradation in ASR performance, especially in languages with a near 1-to-1 grapheme-to-phoneme map. In such cases no lexicon is needed, or at least its creation is trivially the expansion of a word into its constituent letters.

10.2 Problem setup

We assume that we have un-transcribed audio in target language, unrelated text in the same language, and transcribed speech in a separate language. We do not have access to a pronunciation lexicon or a universal phoneme set. The hope is that using either graphemes or a universal phoneme set will result in similar performance. Below are two proposed architectures followed by preliminary results reporting character error rate on both systems. We implement all systems in Kaldi [85].

10.3 System I

A naive solution is to simply train graphemic acoustic models on speech from a resource-rich language and subsequently decode the target language using these graphemic acoustic models. We could in fact train on as many resource-rich languages sharing the same, or almost the same orthography as the target language. In our preliminary experiments, however, we train on a single language. This serves as a baseline system. In our baseline experiment, the Wall Street Journal (WSJ) Corpus is the resource-rich language and the BABEL Turkish corpus is the target language. Clearly there is a severe mismatch between the train and test sets. One way to reduce some of the mismatch is to downsample the WSJ corpus to 8kHz so that both languages have the same bandwidth. In keeping with the WFST framework as proposed in [79], and also used in Kaldi [85], and [135], we look to construct the components $H \circ C \circ L \circ G$ needed for a WFST based ASR system.

For the baseline system we immediately have H, the acoustic model, from the graphemic acoustic models trained on WSJ. We train a context dependent graphemic TDNN system with speed and volume perturbation. Since we are interested in character error rate, we construct a language model on graphemes by simply collapsing all words into a single long grapheme string and training a standard 4-gram language model on this grapheme string.

The knowledge transfer is really cross-lingual lexical transfer where we now assume a 1-to-1 map between Turkish acoustic models and English acoustic models accomplished by simply matching graphemes. If a grapheme does not occur in one language the easiest way to account for it is to 'back off' by removing diacritics the unknown grapheme to a grapheme shared by both languages. Parsing the Unicode description of the grapheme or removing the combining characters when the grapheme is represented in its canonical decomposed form easily accomplishes this backing off procedure. For this experiment (and for most language pairs sharing orthographies) this is sufficient. Should an unknown grapheme with no potential back off grapheme exist in the language, a 1-to-many map can be used for that grapheme such that the unknown grapheme maps to any grapheme from the resource-rich language. Since the trained acoustic models are context dependent models, each Turkish grapheme simply has an alternate pronunciation for each context dependent grapheme in English to which it mapped.

Lastly the following FST is used between C and L to prevent forbidden combinations of context dependent units, (i.e. `H_I E_E L_B L_I O_B`).

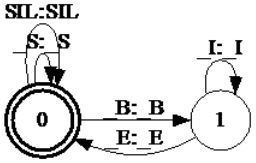


Figure 10.1: FST for Constraining context dependent character sequecnes.

10.4 System II

Having acoustic models in the target language might help relieve the problem of acoustic mismatch, and also to alleviate the assumption of a perfect 1-to-1 map between the target language phonemes and the resource-rich language graphemes. Unfortunately, with no transcribed speech it is impossible to train standard acoustic models. An alternative is to use unsupervised acoustic models. For this setup, we use the Bayesian nonparametric model described in [7]. Once we have discovered acoustic models in the target language we use these in a role similar the Turkers in [135]. In our experiment we generate a ground truth graphemic transcription of speech from the WSJ corpus by force aligning the transcribed text using graphemic acoustic models. In [135] noisy grapheme sequences were transcribed to describe ground truth phoneme sequences whereas here noisy acoustic unit sequences are automatically generated to describe ground-truth grapheme sequences. We simply use the single best path during acoustic unit decoding to form the acoustic unit sequences.

Equipped with aligned pairs of English graphemes and Turkish acoustic unit sequences we then learn a WSFT that accepts acoustic units and emits English graphemes. The transducer is constructed in the following way: For each English grapheme seen in the training corpus, denoted `G_j` below, create the following elementary left-to-right HMM with emissions described by a discrete distribution over the acoustic units denoted `a_i` below. The output `a_i` are to indicate emissions.

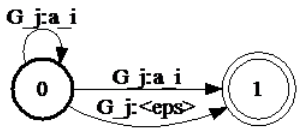


Figure 10.2: Grapheme HMM.

The null arc (with ¡eps¿ emission) allows for 1-to-many mappings of Turkish acoustic units to English graphemes while the self-loop allows for many-to-1 mappings. We then assign acoustic unit segments to the closest grapheme segment as measured by the central frame. From this alignment we estimate the unigram distribution of acoustic units for each grapheme and use these to initialize the emission probabilities for all non-null arcs. We initialize the transition probabilities as 0.8, 0.1, and 0.1 for the 0-1 transition, self-loop and null arc respectively. We then concatenate sequences of HMMs for each utterance to form our HMM topology and perform standard Baum-Welsh training until convergence. We implement add-1 smoothing to account for unseen acoustic units.

This raises a separate issue of how to account for acoustic units seen in English but not in Turkish and visa versa. We consider the inventory of acoustic units to be only those we have previously seen in Turkish since that is the language whose acoustics we are attempting to model. On the other hand if we see a unit in Turkish, but never in English, then the add-1 smoothing takes care of the situation and we do not accidentally eliminate the unit from our acoustic unit inventory. Another interesting alternative would be to weight the emission probabilities for each acoustic unit by the distribution of acoustic units on Turkish.

The above HMM for a single grapheme has an equivalent WFST representation shown below. The symbol `a_i` on an arc represents all the arcs between the same nodes, one for each acoustic unit. G is the grapheme output.

Of course each arc also has a weight. This weight is simply $P(l)P(a\_i|l)$, where l is the arc in the FST. To finally get a general transducer from acoustic units to graphemes, we take union and closure of all the grapheme WFSTs. We will call this component T. We construct L as we did in the baseline system, but ignoring the context dependent graphemes. Composing $T \circ L$, we can consider this new lexicon, LT, a WFST which takes as input acoustic unit sequences and outputs graphemes. One could imagine creating a lexicon for all words, since each word is trivially written in terms of its constituent

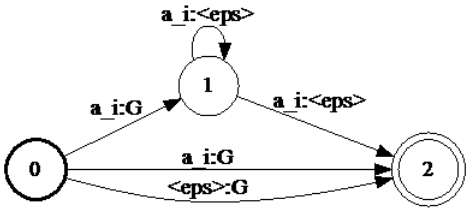


Figure 10.3: Equivalent Grapheme FST.

	CER (Turkish)	PER [135] (CA,MD)
System I (Baseline)	79.6	68.4, 71.3
System II	77.2	57.2, 58.2

Table 10.1: Results.

letters from which we then have a mapping to acoustic units. In this way we have simultaneously discovered and trained acoustic models while simultaneously learning a lexicon.

We then generate acoustic unit lattices for the un-transcribed Turkish audio, the composition of which with LT returns a new WFST that transduces Turkish audio into English graphemes.

10.5 Results and Conclusions

The results of the two experiments described in the previous sections are in table 10.1. In the first column are the character error rates resulting on Turkish from the two systems. In the second column are phone error rates as reported in [135] on the most comparable phonemic systems for the best and worst languages. This is just to give a reference point. It should be noted that the experiments in [135] were run on broadcast news recordings, which are relatively clean speech, where are the BABEL Turkish was noisy telephone speech. This may account for some of the difference. Nonetheless it appears that the system is approaching error rates at which useful ASR tasks are possible

Bibliography

[1] Lukas Burget, Petr Schwarz, et al: "Multilingual acoustic modeling for speech recognition based on Subspace Gaussian Mixture Models." In: *Proc. International Conference on Acoustictics, Speech, and Signal Processing*, 2010.

[2] Patrick Kenny, Douglas Reynolds, and Fabio Castaldo. "Diarization of telephone conversations using factor analysis." *IEEE Journal of Selected Topics in Signal Processing*, 4.6 (2010): 1059-1070.

[3] F. Grezl, E. Egorova, and M. Karafiat: "Further investigation into multilingual training and adaptation of stacked bottle-neck neural network structure", in *Proc. IEEE SLT*, 2014.

[4] Oliver Walter, Reinhold Haeb-Umbach, Sourish Chaudhuri, and Bhiksha Raj, "Unsupervised word discovery from phonetic input using nested Pitman-Yor language modeling", in *Proc. IEEE International Conference on Robotics and Automation (ICRA)*, 2013.

[5] Takafumi Moriya, et al. "Automation of system building for state-of-the-art large vocabulary speech recognition using evolution strategy." in *Proc. IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*, 2015.

[6] Chia-ying Lee and James Glass, "A nonparametric bayesian approach to acoustic model discovery," in *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers - Volume 1*, Stroudsburg, PA, USA, 2012, ACL '12, pp. 40–49, Association for Computational Linguistics.

[7] Lucas Ondel, Lukáš Burget, and Jan Černocký, "Variational inference for acoustic unit discovery," in *Procedia Computer Science*. 2016, vol. 2016, pp. 80–86, Elsevier Science.

[8] Daniel Renshaw, Herman Kamper, Aren Jansen, and Sharon Goldwater, "A comparison of neural network methods for unsupervised representation learning on the zero resource speech challenge," in *INTERSPEECH 2015, 16th Annual Conference of the International Speech Communication Association, Dresden, Germany, September 6-10, 2015*, 2015, pp. 3199–3203.

[9] David M. Blei and Michael I. Jordan, "Variational inference for dirichlet process mixtures," *Bayesian Analysis*, vol. 1, pp. 121–144, 2005.

[10] Charles E. Antoniak, "Mixtures of dirichlet processes with applications to bayesian nonparametric problems," *Annals of Statistics*, vol. 2, no. 6, November 1974.

[11] Carl Edward Rasmussen, "The infinite gaussian mixture model," in *NIPS*, Sara A. Solla, Todd K. Leen, and Klaus-Robert Müller, Eds. 1999, pp. 554–560, The MIT Press.

[12] Yee Whye Teh, "A hierarchical bayesian language model based on pitmanyor processes," in *In Coling/ACL, 2006. 9*, 2006.

[13] Y. W. Teh and M. I. Jordan, "Hierarchical Bayesian nonparametric models with applications," in *Bayesian Nonparametrics: Principles and Practice*, N. Hjort, C. Holmes, P. Müller, and S. Walker, Eds. Cambridge University Press, 2010.

[14] J. S. Garofolo, L. F. Lamel, W. M. Fisher, J. G. Fiscus, D. S. Pallett, and N. L. Dahlgren, "DARPA TIMIT acoustic phonetic continuous speech corpus CDROM," 1993.

[15] S. J. Young, G. Evermann, M. J. F. Gales, T. Hain, D. Kershaw, G. Moore, J. Odell, D. Ollason, D. Povey, V. Valtchev, and P. C. Woodland, *The HTK Book, version 3.4*, Cambridge University Engineering Department, Cambridge, UK, 2006.

[16] František Grézl and Martin Karafiát, "Adapting multilingual neural network hierarchy to a new language," in *Proceedings of the 4th International Workshop on Spoken Language Technologies for Under- resourced Languages SLTU-2014. St. Petersburg, Russia, 2014*. 2014, pp. 39–45, International Speech Communication Association.

[17] Daniel Povey, Arnab Ghoshal, Gilles Boulianne, Lukas Burget, Ondrej Glembek, Nagendra Goel, Mirko Hannemann, Petr Motlicek, Yanmin Qian, Petr Schwarz, Jan Silovsky, Georg Stemmer, and Karel Vesely, "The kaldi speech recognition toolkit," in *IEEE 2011 Workshop on Automatic Speech Recognition and Understanding*. Dec. 2011, IEEE Signal Processing Society, IEEE Catalog No.: CFP11SRW-USB.

[18] Kurihara, K., M. Welling, Y. Teh. Collapsed variational Dirichlet process mixture models. In IJCAI. 2007.

[19] C. Bishop, Pattern Recognition and Machine Learning, vol. 4, Springer, New York, 2006, pp. 461-483.

[20] K. Murphy, Machine Learning: A Probabilistic Perspective. Cambridge, MA: MIT Press, 2012, pp. 842-844.

[21] Laurent Besacier, Etienne Barnard, Alexey Karpov, and Tanja Schultz, "Automatic speech recognition for under-resourced languages: A survey," *Speech Communication*, vol. 56, pp. 85–100, 2014.

[22] Chia-ying Lee and James Glass, "A nonparametric bayesian approach to acoustic model discovery," in *Proc. ACL*, 2012.

[23] Lucas Ondel, Lukaš Burget, and Jan Černockỳ, "Variational inference for acoustic unit discovery," in *Proc. SLTU*, 2016.

[24] Ronald A Fisher, "The use of multiple measurements in taxonomic problems," *Annals of eugenics*, vol. 7, no. 2, pp. 179–188, 1936.

[25] Michael Heck, Sakriani Sakti, and Satoshi Nakamura, "Unsupervised linear discriminant analysis for supporting DPGMM clustering in the zero resource scenario," in *Proc. SLTU*, 2016.

[26] Dong Yu and Michael L Seltzer, "Improved bottleneck features using pretrained deep neural networks.," in *Proc. INTERSPEECH*, 2011.

[27] Karel Veselỳ, Martin Karafiát, František Grézl, Miloš Janda, and Ekaterina Egorova, "The language-independent bottleneck features," in *Proc. SLT*, 2012.

[28] Yaodong Zhang and James R Glass, "Unsupervised spoken keyword spotting via segmental dtw on Gaussian posteriorgrams," in *Proc. ASRU*, 2009.

[29] Hongjie Chen, Cheung-Chi Leung, Lei Xie, Bin Ma, and Haizhou Li, "Parallel inference of dirichlet process Gaussian mixture models for unsupervised acoustic modeling: A feasibility study," in *Proc. INTERSPEECH*, 2015.

[30] Michael Carlin, Samuel Thomas, Aren Jansen, and Hynek Hermansky, "Rapid evaluation of speech representations for spoken term discovery," in *Proc. ICASSP*, 2011.

[31] Aren Jansen, Emmanuel Dupoux, Sharon Goldwater, Mark Johnson, Sanjeev Khudanpur, Kenneth Church, Naomi Feldman, Hynek Hermansky, Florian Metze, Richard Rose, et al., "A summary of the 2012 jhu clsp workshop on zero resource speech technologies and models of early language acquisition," in *Proc. ICASSP*, 2013.

[32] Aren Jansen and Benjamin Van Durme, "Efficient spoken term discovery using randomized algorithms," in *Proc. ASRU*, 2011.

[33] Vince Lyzinski, Gregory Sell, and Aren Jansen, "An evaluation of graph clustering methods for unsupervised term discovery," in *Proc. INTERSPEECH*, 2015.

[34] Man-hung Siu, Herbert Gish, Arthur Chan, William Belfield, and Steve Lowe, "Unsupervised training of an HMM-based self-organizing unit recognizer with applications to topic classification and keyword discovery," *Computer Speech & Language*, vol. 28, no. 1, pp. 210–223, 2014.

[35] Mark Dredze, Aren Jansen, Glen Coppersmith, and Ken Church, "NLP on spoken documents without ASR," in *Proc. EMNLP*, 2010.

[36] David M Blei, Michael I Jordan, et al., "Variational inference for dirichlet process mixtures," *Bayesian analysis*, vol. 1, no. 1, pp. 121–144, 2006.

[37] Michael I Jordan, Zoubin Ghahramani, Tommi S Jaakkola, and Lawrence K Saul, "An introduction to variational methods for graphical models," *Machine learning*, vol. 37, no. 2, pp. 183–233, 1999.

[38] Vijayaditya Peddinti, Daniel Povey, and Sanjeev Khudanpur, "A time delay neural network architecture for efficient modeling of long temporal contexts," in *Proc. INTERSPEECH*, 2015.

[39] Timothy J. Hazen, "MCE Training Techniques for Topic Identification of Spoken Audio Documents," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 19, no. 8, pp. 2451–2460, Nov 2011.

[40] David M. Blei, Andrew Y. Ng, and Michael I. Jordan, "Latent Dirichlet Allocation," *JMLR*, vol. 3, pp. 993–1022, March 2003.

[41] Shai Shalev-Shwartz, Yoram Singer, and Nathan Srebro, "Pegasos: Primal estimated subgradient solver for svm," in *Proceedings of the 24th International Conference on Machine Learning*, New York, NY, USA, 2007, ICML '07, pp. 807–814, ACM.

[42] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[43] George Karypis, "CLUTO: A software package for clustering high-dimensional data sets.," Tech. Rep. 02-017, University of Minnesota, Dept. of Computer Science, 2003.

[44] John Godfrey, Edward Holliman, and Jane McDaniel, "SWITCHBOARD: Telephone speech corpus for research and development," in *Proc. ICASSP*, 1992.

[45] Daniel Povey, Arnab Ghoshal, Gilles Boulianne, Lukas Burget, Ondrej Glembek, Nagendra Goel, Mirko Hannemann, Petr Motlicek, Yanmin Qian, Petr Schwarz, et al., "The Kaldi speech recognition toolkit," in *Proc. ASRU*, 2011.

[46] Amit Bagga and Breck Baldwin, "Entity-based cross-document coreferencing using the vector space model," in *Proceedings of the 17th international conference on Computational linguistics-Volume 1*. Association for Computational Linguistics, 1998, pp. 79–85.

[47] Enrique Amigó, Julio Gonzalo, Javier Artiles, and Felisa Verdejo, "A comparison of extrinsic clustering evaluation metrics based on formal constraints," *Information retrieval*, vol. 12, no. 4, pp. 461–486, 2009.

[48] Herman Kamper, Aren Jansen, and Sharon Goldwater, "A segmental framework for fully-unsupervised large-vocabulary speech recognition," *CoRR*, vol. abs/1606.06950, 2016.

[49] C.Y. Lee and J. Glass, "A nonparametric bayesian approach to acoustic model discovery," in *Proc. of 50th Annual Meeting of the ACL*, Stroudsburg, PA, USA, 2012, pp. 40–49, Association for Computational Linguistics.

[50] Chia-ying Lee, Yu Zhang, and James Glass, "Joint learning of phonetic units and word pronunciations for ASR," in *Proceedings of the 2013 Conference on Empirical Methods on Natural Language Processing (EMNLP)*, 2013, pp. 182–192.

[51] Lucas Ondel, Lukas Burget, and Jan Cernocky, "Variational inference for acoustic unit discovery," SLTU-2016 5th Workshop on Spoken Language Technologies for Under-resourced languages 09-12 May 2016 Yogyakarta, Indonesia, 2016.

[52] Sharon Goldwater, Thomas L. Griffiths, and Mark Johnson, "A bayesian framework for word segmentation: Exploring the effects of context," *Cognition*, vol. 112, no. 1, pp. 21 – 54, 2009.

[53] Daichi Mochihashi, Takeshi Yamada, and Naonori Ueda, "Bayesian unsupervised word segmentation with nested pitman-yor language modeling," in *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 1 - Volume 1*, Stroudsburg, PA, USA, 2009, ACL '09, pp. 100–108, Association for Computational Linguistics.

[54] Graham Neubig, Masato Mimura, Shinsuke Mori, and Tatsuya Kawahara, "Bayesian learning of a language model from continuous speech," *IEICE Transactions on Information and Systems*, vol. E95-D, no. 2, pp. 614–625, February 2012.

[55] Jahn Heymann, Oliver Walter, Reinhold Haeb-Umbach, and Bhiksha Raj, "Unsupervised word segmentation from noisy input," in *Automatic Speech Recognition and Understanding Workshop (ASRU 2013)*, Dec. 2013.

[56] Jahn Heymann, Oliver Walter, Reinhold Haeb-Umbach, and Bhiksha Raj, "Iterative bayesian word segmentation for unsupervised vocabulary discovery from phoneme lattices," in *39th International Conference on Acoustics, Speech and Signal Processing (ICASSP 2014)*, May 2014.

[57] Maarten Versteegh, Roland Thiolliere, Thomas Schatz, Xuan Nga Cao, Xavier Anguera, Aren Jansen, and Emmanuel Dupoux, "The zero resource speech challenge 2015," in *Proceedings of Interspeech*, 2015.

[58] David M. Blei and Michael I. Jordan, "Variational inference for dirichlet process mixtures," *Bayesian Anal.*, vol. 1, no. 1, pp. 121–143, 03 2006.

[59] Yee Whye Teh, "A bayesian interpretation of interpolated kneser-ney," Tech. Rep., NUS School of Computing, 2006.

[60] Maarten Versteegh, Xavier Anguera, Aren Jansen, and Emmanuel Dupoux, "The zero resource speech challenge 2015: Proposed approaches and results," *Procedia Computer Science*, vol. 81, pp. 67 – 72, 2016, SLTU-2016 5th Workshop on Spoken Language Technologies for Under-resourced languages 09-12 May 2016 Yogyakarta, Indonesia.

[61] Thomas Schatz, Vijayaditya Peddinti, Francis Bach, Aren Jansen, Hynek Hermansky, and Emmanuel Dupoux, "Evaluating speech features with the minimal-pair abx task: Analysis of the classical mfc/plp pipeline," in *INTERSPEECH 2013: 14th Annual Conference of the International Speech Communication Association*, 2013, pp. 1–5.

[62] Maximilian Bisani and Hermann Ney, "Joint-Sequence Models for Grapheme-to-Phoneme Conversion," *Speech Communication*, vol. 50, no. 5, pp. 434–451, 2008.

[63] Keigo Kubo, Sakriani Sakti, Graham Neubig, Tomoki Toda, and Satoshi Nakamura, "Structured soft margin confidence weighted learning for grapheme-to-phoneme conversion.," in *Proceedings Interspeech*, 2014, pp. 1263–1267.

[64] Sittichai Jiampojamarn, Colin Cherry, and Grzegorz Kondrak, "Integrating joint n-gram features into a discriminative training framework," in *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the ACL*. 2010, pp. 697–700, Association for Computational Linguistics.

[65] Patrick Lehnen, Alexandre Allauzen, Thomas Lavergne, Francois Yvon, Stefan Hahn, and Hermann Ney, "Structure learning in hidden conditional random fields for grapheme-to-phoneme conversion.," in *Proceedings Interspeech*, 2013, pp. 2326–2330.

[66] Kanishka Rao, Fuchun Peng, Haşim Sak, and Françoise Beaufays, "Grapheme-to-phoneme conversion using long short-term memory recurrent neural networks," in *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2015, pp. 4225–4229.

[67] Kaisheng Yao and Geoffrey Zweig, "Sequence-to-sequence neural net models for grapheme-to-phoneme conversion," *arXiv preprint arXiv:1506.00196*, 2015.

[68] Chia-ying Lee, Yu Zhang, and James R Glass, "Joint learning of phonetic units and word pronunciations for asr.," in *Proceedings EMNLP*, 2013, pp. 182–192.

[69] Josef R Novak, Nobuaki Minematsu, and Keikichi Hirose, "Wfst-based grapheme-to-phoneme conversion: open source tools for alignment, model-building and decoding," in *10th International Workshop on Finite State Methods and Natural Language Processing*, 2012, p. 45.

[70] Josef R Novak, Nobuaki Minematsu, Keikichi Hirose, Chiori Hori, Hideki Kashioka, and Paul R Dixon, "Improving wfst-based g2p conversion with alignment constraints and rnnlm n-best rescoring.," in *Proceedings Interspeech*, 2012, pp. 2526–2529.

[71] Josef R Novak, Nobuaki Minematsu, and Keikichi Hirose, "Failure transitions for joint n-gram models and g2p conversion.," in *Proceedings Interspeech*, 2013, pp. 1821–1825.

[72] Josef Robert Novak, Nobuaki Minematsu, and Keikichi Hirose, "Phonetisaurus: Exploring grapheme-to-phoneme conversion with joint n-gram models in the wfst framework," *Natural Language Engineering*, pp. 1–32, 2015.

[73] Diamantino Caseiro, Isabel Trancoso, Luis Oliveira, and Ceu Viana, "Grapheme-to-phone using finite state transducers," in *Proc. 2002 IEEE Workshop on Speech Synthesis*, 2002, vol. 2, pp. 1349–1360.

[74] Ke Wu, Cyril Allauzen, Keith B Hall, Michael Riley, and Brian Roark, "Encoding linear models as weighted finite-state transducers.," in *Proceedings Interspeech*, 2014, pp. 1258–1262.

[75] Yee Whye Teh, "A hierarchical Bayesian language model based on Pitman-Yor processes," in *Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, 2006, pp. 985–992.

[76] Yee Whye Teh, "A Bayesian interpretation of interpolated Kneser-Ney," Tech. Rep. TRA2/06, School of Computing, 2006.

[77] Sharon Goldwater, Tom Griffiths, and Mark Johnson, "Interpolating between types and tokens by estimating power-law generators," *Advances in neural information processing systems NIPS*, vol. 18, pp. 459, 2006.

[78] Daichi Mochihashi, Takeshi Yamada, and Naonori Ueda, "Bayesian unsupervised word segmentation with nested Pitman-Yor language modeling," in *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 1*. Association for Computational Linguistics, 2009, pp. 100–108.

[79] Mehryar Mohri, Fernando C. N. Pereira, and Michael Riley, "Speech recognition with weighted finite-state transducers," in *Handbook on Speech Processing and Speech Communication, Part E: Speech recognition*, Larry Rabiner and Fred Juang, Eds., Heidelberg, Germany, 2008, p. 31, Springer-Verlag.

[80] Alfred V Aho and Margaret J Corasick, "Efficient string matching: an aid to bibliographic search," *Communications of the ACM*, vol. 18, no. 6, pp. 333–340, 1975.

[81] Jahn Heymann, Oliver Walter, Reinhold Haeb-Umbach, and Bhiksha Raj, "Iterative bayesian word segmentation for unsupervised vocabulary discovery from phoneme lattices," in *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2014, pp. 4057–4061.

[82] J. McDonough, K. Ng, P. Jeanrenaud, H. Gish, and J. R. Rohlicek, "Approaches to topic identification on the switchboard corpus," in *IEEE ICASSP*, Apr 1994, vol. i, pp. I/385–I/388 vol.1.

[83] Timothy J. Hazen, Fred Richardson, and Anna Margolis, "Topic Identification from Audio Recordings using Word and Phone Recognition Lattices," in *IEEE Workshop on ASRU*, December 2007, pp. 659–664.

[84] Vassil Panayotov, Guoguo Chen, Daniel Povey, and Sanjeev Khudanpur, "Librispeech: An ASR Corpus Based on Public Domain Audio Books," in *IEEE ICASSP*, April 2015, pp. 5206–5210.

[85] Daniel Povey, Arnab Ghoshal, Gilles Boulianne, Lukas Burget, Ondrej Glembek, Nagendra Goel, Mirko Hannemann, Petr Motlicek, Yanmin Qian, Petr Schwarz, Jan Silovsky, Georg Stemmer, and Karel Vesely, "The Kaldi Speech Recognition Toolkit," in *IEEE Workshop on ASRU*. Dec 2011, IEEE Signal Processing Society.

[86] Rémi Flamary, Xavier Anguera, and Nuria Oliver, "Spoken WordCloud: Clustering Recurrent Patterns in Speech," in *International Workshop on Content-Based Multimedia Indexing*, June 2011, pp. 133–138.

[87] David F. Harwath, Timothy J. Hazen, and James R. Glass, "Zero Resource Spoken Audio Corpus Analysis," in *IEEE ICASSP*, May 2013, pp. 8555–8559.

[88] Timothy J. Hazen, Man-Hung Siu, Herbert Gish, Steve Lowe, and Arthur Chan, "Topic Modeling for Spoken Documents using only Phonetic Information," in *IEEE Workshop on ASRU*, December 2011, pp. 395–400.

[89] J. Wintrode and S. Khudanpur, "Limited resource term detection for effective topic identification of speech," in *IEEE ICASSP*, May 2014, pp. 7118–7122.

[90] Lucas Ondel, Lukáš Burget, and J. Černocký, "Variational Inference for Acoustic Unit Discovery," *To appear in SLTU*, 2016.

[91] Herbert Gish, Man-Hung Siu, Arthur Chan, and William Belfield, "Unsupervised training of an hmm-based speech recognizer for topic classification," in *INTERSPEECH*, September 2009, pp. 1935–1938.

[92] Man-Hung Siu, Herbert Gish, Arthur Chan, William Belfield, and Steve Lowe, "Unsupervised Training of an HMM-based Self-Organizing Unit Recognizer with Applications to Topic Classification and Keyword Discovery," *Computer Speech & Language*, vol. 28, no. 1, pp. 210–223, 2014.

[93] Léon Bottou and Olivier Bousquet, "The tradeoffs of large scale learning," in *Advances in Neural Information Processing Systems*, J.C. Platt, D. Koller, Y. Singer, and S. Roweis, Eds., vol. 20, pp. 161–168. NIPS Foundation (http://books.nips.cc), 2008.

[94] Karel Veselý, Arnab Ghoshal, Lukás Burget, and Daniel Povey, "Sequence-discriminative training of deep neural networks," in *INTERSPEECH*, August 2013, pp. 2345–2349.

[95] Petr Schwarz, *Phoneme Recognition based on Long Temporal Context*, Ph.D. thesis, Brno University of Technology, 2009.

[96] Martin Karafiát, Frantisek Grézl, Mirko Hannemann, Karel Veselý, and Jan Cernocký, "BUT BABEL system for spontaneous cantonese," in *INTERSPEECH*, August 2013, pp. 2589–2593.

[97] František Grézl and Martin Karafiát, "Adapting multilingual neural network hierarchy to a new language," in *Proceedings of the 4th International Workshop on Spoken Language Technologies for Under- resourced Languages SLTU-2014. St. Petersburg, Russia*, 2014, pp. 39–45.

[98] Tanja Schultz, "Globalphone: a multilingual speech and text database developed at Karlsruhe university," in *7th International Conference on Spoken Language Processing, ICSLP2002 - INTERSPEECH*, September 2002.

[99] R. Zhang, Z.A. Bawab, A. Chan, A. Chotimongkol, D. Huggins-Daines, and A.I. Rudnicky, "Investigations on ensemble based semi-supervised acoustic model training," in *Ninth European Conference on Speech Communication and Technology*. ISCA, 2005.

[100] T. Cincarek, T. Toda, H. Saruwatari, and K. Shikano, "Cost reduction of acoustic modeling for real-environment applications using unsupervised and selective training," *IEICE Transactions on Information and Systems*, vol. E91-D, no. 3, pp. 499–507, 2008.

[101] Karel Veselỳ, Mirko Hannemann, and Lukáš Burget, "Semi-supervised training of deep neural networks," in *Automatic Speech Recognition and Understanding (ASRU), 2013 IEEE Workshop on*. IEEE, 2013, pp. 267–272.

[102] Maximilian Bisani and Hermann Ney, "Joint-sequence models for grapheme-to-phoneme conversion," *Speech Communication*, vol. 50, no. 5, pp. 434–451, 2008.

[103] Stanley F Chen et al., "Conditional and joint models for grapheme-to-phoneme conversion.," in *INTERSPEECH*, 2003.

[104] Paul Taylor, "Hidden Markov models for grapheme to phoneme conversion.," in *INTERSPEECH*, 2005, pp. 1973–1976.

[105] Josef R Novak, Nobuaki Minematsu, and Keikichi Hirose, "WFST-based grapheme-to-phoneme conversion: open source tools for alignment, model-building and decoding," in *10th International Workshop on Finite State Methods and Natural Language Processing*, 2012, p. 45.

[106] Liang Lu, Arnab Ghoshal, and Steve Renals, "Acoustic data-driven pronunciation lexicon for large vocabulary speech recognition," in *Automatic Speech Recognition and Understanding (ASRU), 2013 IEEE Workshop on*. IEEE, 2013, pp. 374–379.

[107] Timothy J Hazen and Issam Bazzi, "A comparison and combination of methods for OOV word detection and word confidence scoring," in *Acoustics, Speech, and Signal Processing, 2001. Proceedings.(ICASSP'01). 2001 IEEE International Conference on*. IEEE, 2001, vol. 1, pp. 397–400.

[108] Carolina Parada, Mark Dredze, Denis Filimonov, and Frederick Jelinek, "Contextual information improves oov detection in speech," in *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*. Association for Computational Linguistics, 2010, pp. 216–224.

[109] Issam Bazzi, *Modelling out-of-vocabulary words for robust speech recognition*, Ph.D. thesis, Massachusetts Institute of Technology, 2002.

[110] Ariya Rastrow, Abhinav Sethy, and Bhuvana Ramabhadran, "A new method for OOV detection using hybrid word/fragment system," in *2009 IEEE International Conference on Acoustics, Speech and Signal Processing*. IEEE, 2009, pp. 3953–3956.

[111] Shinji Watanabe and Jen-Tzung Chien, *Bayesian Speech and Language Processing*, Cambridge University Press, 2015.

[112] M. Mohri, F. Pereira, and M. Riley, "Weighted finite-state transducers in speech recognition," *Computer Speech and Language*, vol. 16, pp. 69–88, 2002.

[113] Y. W. Teh, "Dirichlet processes," in *Encyclopedia of Machine Learning*. Springer, 2010.

[114] Jim Pitman, "Exchangeable and partially exchangeable random partitions," *Probability theory and related fields*, vol. 102, no. 2, pp. 145–158, 1995.

[115] Alan E Gelfand and Adrian FM Smith, "Sampling-based approaches to calculating marginal densities," *Journal of the American statistical association*, vol. 85, no. 410, pp. 398–409, 1990.

[116] J.S. Liu, "The collapsed Gibbs sampler in Bayesian computations with applications to a gene regulation problem," *Journal of the American Statistical Association*, vol. 89, no. 427, 1994.

[117] Yee Whye Teh, "A hierarchical Bayesian language model based on Pitman-Yor processes," in *Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, 2006, pp. 985–992.

[118] Kevin P Murphy, "Hidden semi-Markov models (hsmms)," *unpublished notes*, vol. 2, 2002.

[119] D. Mochihashi, T. Yamada, and N. Ueda, "Bayesian unsupervised word segmentation with nested Pitman-Yor language modeling," in *Proc. ACL-IJCNLP*, 2009, pp. 100–108.

[120] S. Goldwater and T. Griffiths, "A fully Bayesian approach to unsupervised part-of-speech tagging," in *Proc. ACL'07*, 2007, pp. 744–751.

[121] Dan Geiger, Thomas Verma, and Judea Pearl, "Identifying independence in bayesian networks," *Networks*, vol. 20, no. 5, pp. 507–534, 1990.

[122] Graham Neubig, Masato Mimura, Shinsuke Mori, and Tatsuya Kawahara, "Learning a language model from continuous speech.," in *INTERSPEECH*. Citeseer, 2010, pp. 1053–1056.

[123] Graham. Neubig, *Unsupervised learning of lexical information for language processing systems*, Ph.D. thesis, Kyoto University, 2012.

[124] Jahn Heymann, Oliver Walter, Reinhold Haeb-Umbach, and Bhiksha Raj, "Iterative Bayesian word segmentation for unsupervised vocabulary discovery from phoneme lattices," in *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2014, pp. 4057–4061.

[125] J. Garofolo et al., "CSR-I (WSJ0) Complete LDC93S6A. DVD. Philadelphia: Linguistic Data Consortium," 1993.

[126] "CSR-II (WSJ1) Sennheiser LDC94S13B. DVD. Philadelphia: Linguistic Data Consortium," 1994.

[127] Jahn Heymann, Oliver Walter, Reinhold Haeb-Umbach, and Bhiksha Raj, "Unsupervised word segmentation from noisy input," in *Automatic Speech Recognition and Understanding (ASRU), 2013 IEEE Workshop on*. IEEE, 2013, pp. 458–463.

[128] David JC MacKay, *Information theory, inference and learning algorithms*, Cambridge university press, 2003.

[129] Tanja Schultz and Alex Waibel, Language Independent and Language Adaptive Acoustic Modeling for Speech Recognition, *Speech Communication*, vol. 35

[130] Sebastian Stueker, et al. "Integrating multilingual articulatory features into speech recognition.", in *Proc. INTERSPEECH*, 2003.

[131] Pawel Swietojanski, Arnab Ghoshal, and Steve Renals. "Unsupervised cross-lingual knowledge transfer in DNN-based LVCSR." in *Proc. Spoken Language Technology Workshop (SLT)*, 2012.

[132] Jonas Loeoef, Christian Gollan, and Hermann Ney. "Cross-language bootstrapping for unsupervised acoustic model training: rapid development of a Polish speech recognition system.", in *Proc. Interspeech*, 2009.

[133] Tanja Schultz, and Alex Waibel. "Experiments on cross-language acoustic modeling.", in *Proc. INTERSPEECH*, 2001.

[134] Mirjam Killer, Sebastian Stueker, and Tanja Schultz. “Grapheme based speech recognition.”, in *Proc. INTERSPEECH*, 2003.

[135] Chunxi Liu, et al. “Adapting ASR for under-resourced languages using mismatched transcriptions.” in *Proc. 2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2016.