IEICE TRANS. INF. & SYST., VOL.E95–D, NO.2 FEBRUARY 2012

PAPER

Bayesian Learning of a Language Model from Continuous Speech

Graham NEUBIG[†a)], **Masato MIMURA**[†], **Shinsuke MORI**[†], *Nonmembers*, *and* **Tatsuya KAWAHARA**[†], *Member*

SUMMARY We propose a novel scheme to learn a language model (LM) for automatic speech recognition (ASR) directly from continuous speech. In the proposed method, we first generate phoneme lattices using an acoustic model with no linguistic constraints, then perform training over these phoneme lattices, simultaneously learning both lexical units and an LM. As a statistical framework for this learning problem, we use non-parametric Bayesian statistics, which make it possible to balance the learned model's complexity (such as the size of the learned vocabulary) and expressive power, and provide a principled learning algorithm through the use of Gibbs sampling. Implementation is performed using weighted finite state transducers (WFSTs), which allow for the simple handling of lattice input. Experimental results on natural, adult-directed speech demonstrate that LMs built using only continuous speech are able to significantly reduce ASR phoneme error rates. The proposed technique of joint Bayesian learning of lexical units and an LM over lattices is shown to significantly contribute to this improvement.

key words: *language modeling, automatic speech recognition, Bayesian learning, weighted finite state transducers*

1. Introduction

A language model (LM) is an essential part of automatic speech recognition (ASR) systems, providing linguistic constraints on the recognizer and helping to resolve the ambiguity inherent in the acoustic signal. Traditionally, these LMs are learned from digitized text, preferably text that is similar in style and content to the speech that is to be recognized.

In this paper, we propose a new paradigm for LM learning, using not digitized text but audio data of continuous speech. The proposition of learning an LM from continuous speech is motivated from a number of viewpoints. First, the properties of written and spoken language are very different [1], and LMs learned from continuous speech can be expected to naturally model spoken language, removing the need to manually transcribe speech or compensate for these differences when creating an LM for ASR [2]. Second, learning words and their context from speech can allow for out-of-vocabulary word detection and acquisition, which has been shown to be useful in creating more adaptable and robust ASR or dialog systems [3], [4]. Learning LMs from speech could also prove a powerful tool in efforts for technology-based language preservation [5], particularly for languages that have a rich oral, but not written tradition. Finally, as human children learn language from speech, not text, computational models for learning from speech are of great interest in the field of cognitive science [6].

There has been a significant amount of work on learning lexical units from speech data. These include statistical models based on the minimum description length or maximum likelihood frameworks, which have been trained on one-best phoneme recognition results [7]–[9] or recognition lattices [10]. There have also been a number of works that use acoustic matching methods combined with heuristic cutoffs that may be adjusted to determine the granularity of the units that need to be acquired [11]–[13]. Finally, many works, inspired by the multi-modal learning of human children, use visual and audio information (or at least abstractions of such) to learn words without text [6], [14], [15].

This work is different from these other approaches in that it is the first model that is able to learn a full word-based n-gram model from raw audio. In order to learn an LM from continuous speech, we first generate lattices of phonemes without any linguistic constraints using a standard ASR acoustic model. To learn an LM from this data, we build on recent work in unsupervised word segmentation of text [16], proposing a novel inference procedure that allows for models to be learned over lattice input. For LM learning, we use the hierarchical Pitman-Yor LM (HPYLM) [17], a variety of LM that is based on non-parametric Bayesian statistics. Non-parametric Bayesian statistics are well suited to this learning problem, as they allow for automatically balancing model complexity and expressiveness, and have a principled framework for learning through the use of Gibbs sampling.

To perform sampling over phoneme lattices, we represent all of our models using weighted finite state transducers (WFSTs), which allow for simple and efficient combination of the phoneme lattices with the LM. Using this combined lattice, we use a variant of the forward-backward algorithm to efficiently sample a phoneme string and word segmentation according to the model probabilities. By performing this procedure on each of the utterances in the corpus for several iterations, it is possible to effectively discover phoneme strings and lexical units appropriate for LM learning, even in the face of acoustic uncertainty.

In order to evaluate the feasibility of the proposed method, we performed an experiment on learning an LM from only audio files of fluent adult-directed meeting speech with no accompanying text. We demonstrate that, despite the lack of any text data, the proposed model is able to both decrease the phoneme recognition error rate over a separate test set and acquire a lexicon with many intuitively reason-

Manuscript received June 21, 2011.
Manuscript revised September 26, 2011.

[†]The authors are with the Graduate School of Informatics, Kyoto University, Kyoto-shi, 606–8501 Japan.
a) E-mail: neubig@ar.media.kyoto-u.ac.jp

DOI: 10.1587/transinf.E95.D.614

able lexical entries. Moreover, we demonstrate that the proposed lattice processing approach is effective for overcoming acoustic ambiguity present during the training process.

In Sect. 2 we briefly overview ASR, including language modeling and representation of ASR models in the WFST framework. Section 3 describes previous research on LM-based unsupervised word segmentation, which learns LMs even when there are no clear boundaries between words. In Sect. 4 we propose a method for formulating LM-based unsupervised word segmentation using a combination of WFSTs and Gibbs sampling. We conclude the description in Sect. 4.3 by showing that the WFST-based formulation allows for LM learning directly from speech, even in the presence of acoustic uncertainty. Section 5 describes the results of an experimental evaluation demonstrating the effectiveness of the proposed method, and Sect. 6 concludes the paper and discusses future directions.

2. Speech Recognition and Language Modeling

This section provides an overview of ASR and language modeling and provides definitions that will be used in the rest of the paper.

2.1 Speech Recognition

ASR can be formalized as the task of finding a series of words W given acoustic features X of a speech signal containing these words. Most ASR systems use statistical methods, creating a model for the posterior probability of the words given the acoustic features, and searching for the word sequence that maximizes this probability

$$\hat{W} = \underset{W}{\operatorname{argmax}} P(W|X). \tag{1}$$

As this posterior probability is difficult to model directly, Bayes's law is used to decompose the probability

$$\hat{W} = \underset{W}{\operatorname{argmax}} \frac{P(X|W)P(W)}{P(X)} \tag{2}$$

$$= \underset{W}{\operatorname{argmax}} P(X|W)P(W). \tag{3}$$

Here, $P(X|W)$ is computed by the acoustic model (AM), which makes a probabilistic connection between words and their acoustic features. However, directly modeling the acoustic features of the thousands to millions of words in large-vocabulary ASR systems is not realistic due to data sparsity issues. Instead, AMs are trained to recognize sequences of phonemes Y, which are then mapped into the word sequence W. Phonemes are defined as the smallest perceptible linguistic unit of speech. Thus, the entire ASR process can be described as finding the optimal word sequence according to the following formula

$$\hat{W} = \underset{W}{\operatorname{argmax}} \sum_{Y} P(X|Y)P(Y|W)P(W). \tag{4}$$

This is usually further approximated by choosing the single most likely phoneme sequence to allow for efficient search:

$$\hat{W} = \underset{W,Y}{\operatorname{argmax}} P(X|Y)P(Y|W)P(W). \tag{5}$$

Here, $P(X|Y)$ indicates the AM probability and $P(Y|W)$ is a lexicon probability that maps between words and their pronunciations. $P(W)$ is computed by the LM, which we will describe in more detail in the following section. It should be noted that in many cases a scaling factor α is used

$$\hat{W} = \underset{W,Y}{\operatorname{argmax}} P(X|Y)P(Y|W)P(W)^{\alpha}. \tag{6}$$

This allows for the adjustment of the relative weight put on the LM probability.

2.2 Language Modeling

The goal of the LM probability $P(W)$ is to provide a preference towards "good" word sequences, assigning high probability to word sequences that the speaker is likely to say, and low probability to word sequences that the speaker is unlikely to say. By doing so, this allows the ASR system to select linguistically proper sequences when purely acoustic information is not enough to correctly recognize the input.

The most popular form of LM is the n-gram, which is notable for its simplicity, computational efficiency, and surprising power [18]. n-gram LMs are based on the fact that it is possible to calculate the joint probability of $W = w_1^I$ sequentially by conditioning on all previous words in the sequence using the chain rule

$$P(W) = \prod_{i=1}^{I} P(w_i|w_1^{i-1}). \tag{7}$$

Conditioning on previous words in the sequence allows for the consideration of contextual information in the probabilistic model. However, as few sentences will contain exactly the same words as any other, conditioning on all previous words in the sentence quickly leads to data sparseness issues. n-gram models resolve this problem by only conditioning on the previous $(n-1)$ words when choosing the next word in the sequence

$$P(W) \approx \prod_{i=1}^{I} P(w_i|w_{i-n+1}^{i-1}). \tag{8}$$

The conditional probabilities are generally trained from a large corpus of word sequences $\mathcal{W}$. From $\mathcal{W}$ we calculate the counts of each subsequence of n words w_{i-n+1}^{i} (an "n-gram"). From these counts, it is possible to compute conditional probabilities using maximum likelihood estimation

$$P_{ml}(w_i|w_{i-n+1}^{i-1}) = \frac{c(w_{i-n+1}^{i})}{c(w_{i-n+1}^{i-1})}. \tag{9}$$

However, even if we set n to a relatively small value, we will never have a corpus large enough to exhaustively cover

all possible n-grams. In order to deal with this data sparsity issue, it is common to use a framework that references higher order n-gram probabilities when they are available, and falls back to lower order n-gram probabilities according to a *fallback* probability $P(FB|w_{i-n+1}^{i-1})$:

$$P(w_i|w_{i-n+1}^{i-1}) = \begin{cases} P_s(w_i|w_{i-n+1}^{i-1}) & \text{if } c(w_{i-n+1}^{i}) > 0, \\ P(FB|w_{i-n+1}^{i-1})P(w_i|w_{i-n+2}^{i-1}) & \text{otherwise.} \end{cases} \tag{10}$$

By combining more accurate but sparse higher-order n-grams with less accurate but more reliable lower-order n-grams, it is possible to create LMs that are both accurate and robust. To reserve some probability for $P(FB|w_{i-n+1}^{i-1})$, we replace P_{ml} with the smoothed probability distribution P_s. P_s can be defined according to a number of smoothing methods, which are described thoroughly in [19].

2.3 Bayesian Language Modeling

While traditional methods for LM smoothing are based on heuristics (often theoretically motivated), it is also possible to motivate language modeling from the perspective of Bayesian statistics [17], [20]. In order to perform smoothing in the Bayesian framework, we first define a variable $g_{w_i|w_{i-m+1}^{i-1}}$ that specifies n-gram probabilities

$$g_{w_i|w_{i-m+1}^{i-1}} = P(w_i|w_{i-m+1}^{i-1}) \tag{11}$$

where $0 \leq m \leq n-1$ is the length of the context being considered.

As we are not sure of the actual values of the n-gram probabilities due to data sparseness, the standard practice of Bayesian statistics suggests we treat all probabilities as random variables G that we can learn from the training data $\mathcal{W}$. Formally, this learning problem consists of estimating the posterior probability $P(G|\mathcal{W})$. This can be calculated in a Bayesian fashion by placing a prior probability $P(G)$ over G and combining this with the likelihood $P(\mathcal{W}|G)$ and the evidence $P(\mathcal{W})$

$$P(G|\mathcal{W}) = \frac{P(\mathcal{W}|G)P(G)}{P(\mathcal{W})} \tag{12}$$

$$\propto P(\mathcal{W}|G)P(G). \tag{13}$$

We can generally ignore the evidence probability, as the training data is fixed throughout the entire training process.

It should be noted that LMs are a collection of multinomial distributions $G_{w_{i-m+1}^{i-1}} = \{g_{w_i=1|w_{i-m+1}^{i-1}}, \ldots, g_{w_i=N|w_{i-m+1}^{i-1}}\}$ where N is the number of words in the vocabulary. There is one multinomial for each history w_{i-m+1}^{i-1}, with the length of w_{i-m+1}^{i-1} being 0 through $n-1$. As the variables in $G_{w_{i-m+1}^{i-1}}$ belong to a multinomial distribution, it is natural to use priors based on the Pitman-Yor process [21].† The Pitman-Yor process is useful in that it is able to assign probabilities to the space of variables that form multinomial distributions. Formally, this means that if we define the prior over $G_{w_{i-m+1}^{i-1}}$ using a Pitman-Yor process, we will be guaranteed that its elements will add to one

$$\sum_{x=1}^{N} g_{w_i=x|w_{i-m+1}^{i-1}} = 1 \tag{14}$$

and be between zero and one

$$\forall_{x=1}^{N} 0 \leq g_{w_i=x|w_{i-m+1}^{i-1}} \leq 1. \tag{15}$$

The Pitman-Yor process has three parameters: the discount parameter d_m, the strength parameter θ_m, and the base measure $G_{w_{i-m+2}^{i-1}}$

$$G_{w_{i-m+1}^{i-1}} \sim PY(d_m, \theta_m, G_{w_{i-m+2}^{i-1}}). \tag{16}$$

The discount d_m is subtracted from observed counts, and when it is given a large value (close to one), the model will give more probability to frequent words. The strength θ_m controls the overall sparseness of the distribution, and when it is given a small value the distribution will be sparse.†† The base measure $G_{w_{i-m+2}^{i-1}}$ of the Pitman-Yor process indicates the expected value of the probability distributions it generates, and is essentially the "default" value used when there are no words in the training corpus for context w_{i-m+1}^{i-1}.

It should be noted that here, we are setting the base measure of each $G_{w_{i-m+1}^{i-1}}$ to that of its parent context $G_{w_{i-m+2}^{i-1}}$. This forms a hierarchical structure that is referred to as the hierarchical Pitman-Yor LM (HPYLM, [17]) and shown in Fig. 1. This hierarchical structure implies that each set of m-gram (e.g., trigram) probabilities will be using its corresponding $(m-1)$-gram (e.g., bigram) probabilities as a starting point when no or little training data is available. As a result, we achieve a principled probabilistic interpolation of m-gram and $(m-1)$-gram smoothing similar to the heuristic methods described in Sect. 2.2. Finally, the base measure of the unigram model G_0 indicates the prior probability over words in the vocabulary. If we have a vocabulary of all the words that the HPYLM is expected to generate, we can simply set this so that a uniform probability is given to each word in the vocabulary.

For the Pitman-Yor process, the actual probabilities of the LM can be calculated through Gibbs sampling and the Chinese Restaurant Process (CRP) formulation, the details

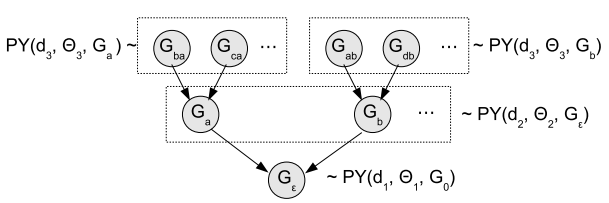


Fig. 1 An example of the hierarchical structure of the HPYLM.

†The better-known Dirichlet process is a specific case of the Pitman-Yor process, where the discount parameter is set to zero.

††Following [17], we give the strength and discount parameters a prior and allow them to be chosen automatically.

of which are beyond the scope of this paper but described in [17]. The important thing to note is that for each n-gram probability, it is possible to calculate the expectation of the probability given a set of sufficient statistics S

$$P(w_i|w_{i-n+1}^{i-1}, S) = \int_0^1 g_{w_i|w_{i-n+1}^{i-1}} P(g_{w_i|w_{i-n+1}^{i-1}}|S) dg_{w_i|w_{i-n+1}^{i-1}}. \quad (17)$$

The statistics S mainly consist of n-gram counts, but also some auxiliary variables that summarize the configuration of the CRP. These can be easily computed given a word-segmented corpus $\mathcal{W}$. The practical implication of this is that we do not need to directly estimate the parameters G, but only need to keep track of the sufficient statistics needed to calculate this expectation of $P(w_i|w_{i-n+1}^{i-1}, S)$. This fact becomes useful when using this model in unsupervised learning, as described in later sections.

2.4 Weighted Finite State ASR

In recent years, the paradigm of weighted finite state transducers (WFSTs) has brought about great increases in the speed and flexibility of ASR systems [22]. Finite state transducers are finite automata with transitions labeled with input and output symbols. WFSTs also assign a weight to transitions, allowing for the definition of weighted relations between two strings. These weights can be used to represent probabilities of each model for ASR including the AM, lexicon, and the LM, examples of which are shown in Fig. 2. In figures of the WFSTs, edges are labeled as "a/b:c", where a indicates the input, b indicates the output, and c indicates the weight. b may be omitted when a and b are the same value, and c will be omitted when it is equal to 1.

The standard AM for $P(X|Y)$ in most ASR systems is based on a Hidden Markov Model (HMM), and its WFST representation, which we will call A. A simplified example of this model is shown in Fig. 2 (a). As input, this takes acoustic features, and after several steps through the HMM outputs a single phoneme such as "e-" or "s." The transition and emission probabilities are identical to the standard HMM used in ASR acoustic models, but we have omitted them from the figure for simplicity.

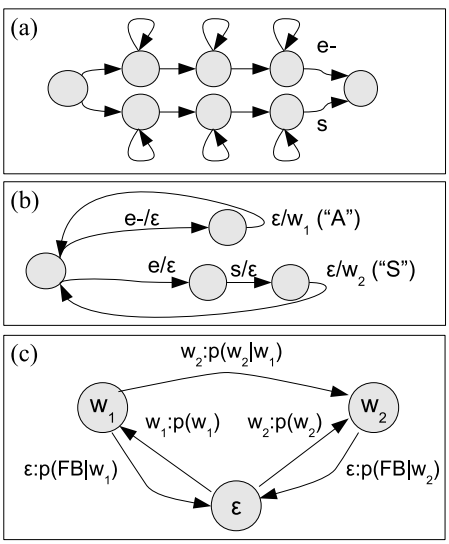


Fig. 2 The WFSTs for ASR including (a) the acoustic model A, (b) the lexicon L, and (c) the language model G.

The WFST formulation for the lexicon, which we will call L, shown in Fig. 2 (b), takes phonemes as input and outputs words along with their corresponding lexicon probability $P(Y|W)$. Excluding the case of homographs (words with the same spelling but different pronunciations), the probability of transitions in the lexicon will be 1.

Finally, the LM probability $P(W)$ can also be represented in the WFST format. Figure 2 (c) shows an example of a bigram LM with only two words w_1 and w_2 in the vocabulary. Each node represents a unique n-gram context w_{i-m+1}^{i-1}, and the outgoing edges from the node represent the probability of symbols given this context $P(w_i|w_{i-m+1}^{i-1})$. In order to handle the fallback to lower-order contexts as described in Sect. 2.2, edges that fall back from w_{i-m+1}^{i-1} to w_{i-m+2}^{i-1} are added, weighted with the fallback probability (marked with "FB" in the figure). The label ϵ on these edges indicates the empty string, which means they can be followed at any time, regardless of the input symbol.

The main advantage of using WFSTs to describe the ASR problem is the existence of efficient algorithms for operations such as composition, intersection, determinization, and minimization. In particular, composition (written $X \circ Y$) allows the combination of two WFSTs in sequence, so if we compose $A \circ L \circ G$ together, we can create a single WFST that takes acoustic features as input and outputs weighted strings of words entailed by the acoustic features. We use this property of WFSTs later to facilitate the implementation of our learning of LMs from continuous speech.

3. Learning LMs from Unsegmented Text

While Sects. 2.2 and 2.3 described how to learn LMs when we are given a corpus of word sequences $\mathcal{W}$, there are some cases when the word sequence is not obvious. For example, when human babies learn words they do so from continuous speech, even though there often are not explicit boundaries between words in the phoneme stream. In addition, many languages such as Japanese and Chinese are written without boundaries between words, and thus the definition of words is not uniquely fixed. These two facts have led to significant research interest in unsupervised word segmentation (WS), the task of finding words and learning LMs from unsegmented phoneme or character strings with no manual intervention [7], [16], [23]–[26].

3.1 Unsupervised WS Modeling

In this work, we follow [16] in taking an LM-based approach to unsupervised WS, learning a word-based LM G from a corpus of unsegmented phoneme strings $\mathcal{Y}$. This

problem can be specified as finding a model according to the posterior probability of the LM $P(G|\mathcal{Y})$, which we can decompose using Bayes's law

$$P(G|\mathcal{Y}) \propto P(\mathcal{Y}|G)P(G). \tag{18}$$

However, as G is a word-based LM, we also assume that there are hidden word sequences $\mathcal{W}$, and model the probability given these sequences

$$P(G|\mathcal{Y}) \propto \sum_{\mathcal{W}} P(\mathcal{Y}|\mathcal{W})P(\mathcal{W}|G)P(G). \tag{19}$$

Here, $P(\mathcal{Y}|\mathcal{W})$ indicates that the words in $\mathcal{W}$ must correspond to the phonemes in $\mathcal{Y}$, and will be 1 if and only if $\mathcal{Y}$ can be recovered by concatenating the words in $\mathcal{W}$ together. $P(\mathcal{W}|G)$ is the likelihood given the LM probabilities, and is identical to that described in Eq. (8).

$P(G)$ can be set using the previously described HPYLM, with one adjustment. With the model we described in Sect. 2.3, it was necessary to know the full vocabulary in advance so that we could set the base measure G_0 to a uniform distribution over all the words in the vocabulary. However, when learning an LM from unsegmented text, W is not known in advance, and thus it is impossible to define a closed vocabulary before training starts. As a result, it is necessary to find an alternative method of defining G_0 that allows the model to flexibly decide which words to include in the vocabulary as training progresses.

In order to do so, [16] uses a "spelling model" H, which assigns prior probabilities over words by using an LM specified over phonemes. If we have a word w_i that consists of phonemes, $y_1, \ldots, y_J$, we define the spelling model probability of w_i according to the n-gram probabilities of H:

$$G_0(w_i) = P(w_i = y_1, \ldots, y_J|H) = \prod_{j=1}^{J} H_{y_j|y_{j-n+1}^{j-1}} \tag{20}$$

We assume that H is also distributed according to the HPYLM, and that the set of phonemes is closed and thus we are able to define a uniform distribution over phonemes H_0. The probabilities of H can be calculated from the set of phoneme sequences of words generated from the spelling model, much like the probabilities of G can be calculated from the set of word sequences contained in the corpus.

This gives us a full generative model for the corpus $\mathcal{Y}$ that first generates the LM probabilities

$$H \sim HPYLM(\boldsymbol{d}_H, \boldsymbol{\theta}_H, H_0) \tag{21}$$

$$G \sim HPYLM(\boldsymbol{d}_G, \boldsymbol{\theta}_G, P(w|H)) \tag{22}$$

then generates each word sequence $W \in \mathcal{W}$ and concatenates it into a phoneme sequence

$$W \sim P(W|G) \tag{23}$$

$$Y \leftarrow \text{concat}(W). \tag{24}$$

This generative story is important in that it allows for the creation of LMs that are both highly expressive and compact (and thus have high generalization capacity). The HPYLM priors for H and G have a preference for simple models, and thus will tend to induce compact models, while the likelihoods for W bias towards larger and more expressive models that describe the data well.

3.2 Inference for Unsupervised WS

The main difficulty in learning LM G from the phoneme string $\mathcal{Y}$ is solving Eq. (19). Here, it is necessary to sum over all possible configurations of $\mathcal{W}$, which represent all possible segmentations of $\mathcal{Y}$. However, for all but the smallest of corpora, the number of possible segmentations is astronomical and thus it is impractical to explicitly enumerate all possible $\mathcal{W}$.

Instead, we can turn to Gibbs sampling [27], [28], a method for calculating this sum approximately. Gibbs sampling approximates the integral or sum over multivariate distributions by stepping through each variable in the distribution and sampling it given all of the other variables to be estimated. As we are interested in calculating $\mathcal{W}$, for each step of the algorithm we take a single sentence $W_k \in \mathcal{W}$ and sample it according to a distribution $P(W_k|Y_k, S_{-W_k})$. S indicates the sufficient statistics calculated from the current configuration of $\mathcal{W}$ required to calculate language model probabilities (as described in Sect. 2.3). S_{-W_k} indicates the sufficient statistics after subtracting the n-gram counts and corresponding CRP configurations that were obtained from the sentence W_k.[†] These sufficient statistics allow us to calculate the conditional probability of W_k given all other sentences, a requirement to properly perform Gibbs sampling. It should be noted that each W_k contains multiple variables (words), so this is a variant of "blocked Gibbs sampling," which samples multiple variables simultaneously [29]. The full sampling procedure is shown in Fig. 3, and we further detail how a single sentence W_k can be sampled according to this distribution in the following section.

By repeating Gibbs sampling for many iterations, the sampled values of each sentence W_k, and the LM sufficient statistics S calculated therefrom, will gradually approach the high-probability areas specified by the model. As men-

Input: Unsegmented corpus $\mathcal{Y}$
Output: Word segmented corpus $\mathcal{W}$
  for all Iterations i in $\{1, \ldots, I\}$ **do**
    for all Sentence k in $\{1, \ldots, |\mathcal{Y}|\}$ **do**
      if $i \neq 1$ **then**
        Remove sufficient statistics obtained from W_k from S
      end if
      Sample a new value of W_k from $P(W_k|Y_k, S_{-W_k})$
      Add the sufficient statistics of the new W_k back to S
    end for
    Save a sample S_i and $\mathcal{W}_i$
  end for

Fig. 3 The algorithm for Gibbs sampling of the word sequence $\mathcal{W}$ and the sufficient statistics S necessary for calculating LM probabilities.

[†]On the first iteration, we start with an empty S, and gradually add the statistics for each sentence as they are sampled.

tioned previously, the HPYLM-based formulation prefers highly expressive, compact models. Lexicons that contain many words are penalized by the HPYLM prior, preventing segmentations of $\mathcal{W}$ that result in a large number of unique words. On the other hand, if the lexicon is too small, it will result in low descriptive power. Thus the sampled values are expected to be those with a consistent segmentation for words, and with common phoneme sequences grouped together as single words.

3.3 Calculating Predictive Probabilities

As the main objective of an LM is to assign a probability to an unseen phoneme string Y, we are interested in calculating the predictive distribution

$$P(Y|\mathcal{Y}) = \int_G \sum_{W\in\{\tilde{W}:\text{concat}(\tilde{W})=Y\}} P(W|G)P(G|\mathcal{Y})dG. \quad (25)$$

However, computing this function directly is computationally difficult. To reduce this computational load we approximate the summation over W with the maximization, assuming that the probability of Y is equal to that of its most likely segmentation.

In addition, assume we have I effective samples of the sufficient statistics obtained after iterations of the previous sampling process.[†] Using these samples, we can approximate the integral over G with the mean of the probabilities given the sufficient statistics $\{S_1, \ldots, S_I\}$

$$P(Y|\mathcal{Y}) \approx \frac{1}{I}\sum_{i=1}^{I} \max_{W\in\{\tilde{W}:\text{concat}(\tilde{W})=Y\}} P(W|S_i). \quad (26)$$

While Eq. (26) approximates the probability using the average maximum-segmentation probability of each S_i, search for such a solution at decoding time is a non-trivial problem. As an approximation to this sum, we find the one-best solution mandated by each of the samples, and combine the separate solutions using ROVER [30].

4. WFST-based Sampling of Word Sequences

While the previous section described the general flow of the inference process, we still require an effective method to sample the word sequence W according to the probability $P(W|Y, S_{-W})$. One way to do so would be to explicitly enumerate all possible segmentations for Y, calculate their probabilities, and sample based on these probabilities. However, as the number of possible segmentations of Y grows exponentially in the length of the sentence, this is an unrealistic solution. Thus, the most difficult challenge of the algorithm in Fig. 3 is efficiently obtaining a word sequence W given a phoneme sequence Y according to the language model probabilities specified by S_{-W}.

One solution is proposed by [16], who use a dynamic programming algorithm that allows for efficient sampling of a value for W according to the probability $P(W|Y, S_{-W})$. While this method is applicable to unsegmented text strings, it is not applicable to situations where uncertainty exists in the input, such as the case of learning from speech. Here we propose an alternative formulation that uses the WFST framework. This is done by first creating a WFST-based formulation of the WS model (Sect. 4.1), then describing a dynamic programming method for sampling over WFSTs (Sect. 4.2). This formulation is critical for learning from continuous speech, as it allows for sampling a word string W from not only one-best phoneme strings, but also phoneme lattices that are able to encode the uncertainty inherent in acoustic matching results.

4.1 A WFST Formulation for Word Segmentation

Our formulation for sampling word sequences consists of first generating a lattice of all possible segmentation candidates using WFSTs, then performing sampling over this lattice. The three WFSTs used for WS (Fig. 4) are quite similar to the ASR WFSTs shown in Fig. 2.

In place of the acoustic model WFST used in ASR, we

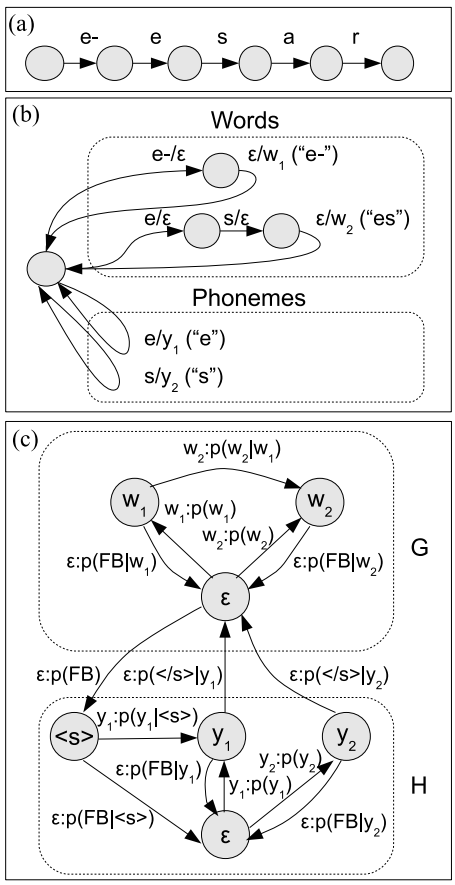


Fig. 4 The WFSTs for word segmentation including (a) the input Y, (b) the lexicon L, and (c) the language model GH.

[†]Some samples may be skipped during the early stages of sampling (a process called "burn-in") to help ensure that samples are likely according to the HPYLM.

simply use a linear chain representing the phonemes in Y, as shown in Fig. 4 (a). The lexicon WFST L in Fig. 4 (b) is identical to the lexicon WFST used in ASR, except that in addition to creating words from phonemes, it also allows all phonemes in the input to be passed through as-is. This allows words in the lexicon to be assigned word-based probabilities according to the language model G, and all words (in the lexicon or not) to be assigned probabilities according to the spelling model H. This is important in the unsupervised WS setting, where the lexicon is not defined in advance, and words outside of the lexicon are still assigned a small probability.

The training process starts with an empty lexicon, and thus no paths emitting words are present. When a word that is not in the lexicon is sampled as a phoneme sequence, L is modified by adding a path that converts the new word's phonemes into its corresponding word token. Conversely, when the last sample containing a word in the lexicon is subtracted from the distribution and the word's count becomes zero, its corresponding path is removed from L. It should be noted that we assume that each word can be mapped onto a single spelling, so $P(Y|W)$ will always be 1.[†]

More major changes are made to the LM WFST, which is shown in Fig. 4 (c). Unlike the case in ASR, where we are generally only concerned with words that exist in the vocabulary, it is necessary to model unknown words that are not included in the vocabulary. The key to the representation is that the word-based LM G and the phoneme-based spelling model H are represented in a single WFST, which we will call GH. GH has weighted edges falling back from the base state of G to H, and edges accepting the terminal symbol for unknown words and transitioning from H to the base state of G. This allows for the WFST to transition as necessary between the known word model and the spelling model.

By composing together these three WFSTs as $Y \circ L \circ GH$, it is possible to create a WFST representing a lattice of segmentation candidates weighted with probabilities according to the LM.

4.2 Sampling over WFSTs

Once we have a WFST lattice representing the model probabilities, we can sample a single path through the WFST according to the probabilities assigned to each edge. This is done using a technique called *forward-filtering/backward-sampling*, a concept similar to that of the forward-backward algorithm for hidden Markov models (HMM). This algorithm can be used to acquire a sample from all probabilistically weighted, acyclic WFSTs defined by a set of states S and a set of edges E.

The first step of the algorithm consists of choosing an ordering for the states in S, which we will write $s_1, \ldots, s_I$. This ordering must be chosen so that all states included in paths that travel to state s_i should be processed before s_i itself. Each edge in E is defined as $e_k = \langle s_i, s_j, w_k \rangle$ traveling from s_i to s_j and weighted by w_k. Assuming the graph is acyclic, we can choose the ordering so that for all edges in E, $i < j$. Given this ordering, if all states are processed in ascending order, we can be ensured that all states will be processed after their predecessors.

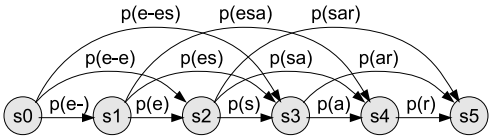


Fig. 5 A WFSA representing a unigram segmentation (words of length greater than three are not displayed).

Next, we perform the forward filtering step, identical to the forward pass of the forward-backward algorithm for HMMs, where probabilities are accumulated from the start state to following states. The initial state s_0 is given a forward probability $f_0 = 1$, and all following states are updated with the sum of the forward probabilities of each of the incoming states multiplied by the weights of the edges to the current state

$$f_j = \sum_{e_k = \langle s_i, s_j, w_k \rangle \in \{E : \tilde{j} = j\}} f_i * w_k. \tag{27}$$

This forward probability can be interpreted as the total probability of all paths that travel to f_j from the initial state.

We provide an example of this process using a weighted finite state acceptor (WFSA) for the unigram segmentation model of "e- e s a r" ("ASR") shown in Fig. 5. In this case, the forward step will push probabilities from the first state as follows:

$$f_1 = P(\text{e-}) * f_0 \tag{28}$$

$$f_2 = P(\text{e-e}) * f_0 + P(\text{e}) * f_1 \tag{29}$$

$$\vdots$$

The backward sampling step of the algorithm consists of sampling a path starting at the final state s_I of the WFST. For the current state, s_j, we can calculate the probability of all incoming edges

$$P(e_k = \langle s_i, s_j, w_k \rangle) = \frac{f_i * w_k}{f_j}, \tag{30}$$

and sample a single incoming edge according to this probability. Here w_k considers the likelihood of e_k itself, while f_i considers the likelihood of all paths traveling up to s_i, allowing for the correct sampling of an edge e_k according to the probability of all paths that travel through it to the current state s_j. In the example, the edge incoming to state s_5 is sampled according to

$$P(s_4 \rightarrow s_5) = P(\text{r}) * f_4 \tag{31}$$

$$P(s_3 \rightarrow s_5) = P(\text{ar}) * f_3 \tag{32}$$

$$\vdots$$

[†]In this work, we assume that all words are represented by their phonetic spelling, not considering the graphemic representation used in usual text. For example, the word "ASR" will be transcribed as "e-esar" in the learned model.

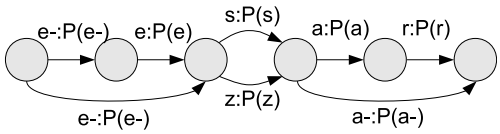


Fig. 6 A WFSA representing a phoneme lattice.

Through this process, a path representing the segmentation of the phoneme string can be sampled according to the probability of the models included in the lattice. Given this path, it is possible to recover Y and W by concatenating the phonemes and words represented by the input and output of the sampled path respectively.

4.3 Extension to Continuous Speech Input

When learning from continuous speech, the input is not a set of phoneme strings $\mathcal{Y}$, but a set of spoken utterances $\mathcal{X}$. As a result, instead of sampling just the word sequences $\mathcal{W}$, we now need to additionally sample the phoneme strings $\mathcal{Y}$. If we can create a single lattice representing the probability of both W and Y for a particular X, it is possible to use the forward-filtering/backward-sampling algorithm to sample phoneme strings and their segmentations together.

With the WFST-based formulation described in the previous section, it is straight-forward to create this lattice representing candidates for Y and W. In fact, all we must do is replace the string of phonemes Y that was used in the WS model in Fig. 4 (a) with the acoustic model HMM A used for ASR in Fig. 2. As a result, the composed lattice $A \circ L \circ GH$ can take acoustic features as input, and includes both the acoustic and language model probabilities. Using this value, we can sample appropriate new values of Y and W, and plug this into the learning algorithm of Fig. 3.

However, as with traditional ASR, if we simply expand all hypotheses allowed by the acoustic model during the forward-filtering step, the hypothesis space will grow unmanageably large. As a solution to this, before starting training we first perform ASR using only the acoustic model and no linguistic information, generating trimmed phoneme lattices representing candidates for each Y such as those shown in Fig. 6.

It should be noted that this dependence on an acoustic model to estimate $P(X|Y)$ indicates that this is not an entirely unsupervised method. However, some work has been done on language-independent acoustic model training [31], as well as the unsupervised discovery and clustering of acoustic units from raw speech [32]. The proposed LM acquisition method could be used in combination with these AM acquisition methods to achieve fully unsupervised speech recognition, a challenge that we leave to future work.

5. Experimental Evaluation

We evaluated the feasibility of the proposed method on continuous speech from meetings of the Japanese Diet (Parliament). This was chosen as an example of naturally spoken, interactive, adult-directed speech with a potentially large vocabulary, as opposed to the simplified grammars or infant-directed speech used in some previous work [6], [14].

5.1 Experimental Setup

We created phoneme lattices using a triphone acoustic model, performing decoding with a vocabulary of 385 syllables that represent the phoneme transitions allowed by the syllable model.[†] No additional linguistic information was used during the creation of the lattices, with all syllables in the vocabulary being given a uniform probability.

In order to assess the amount of data needed to effectively learn an LM, we performed experiments using five different corpora of varying sizes: 7.9, 16.1, 31.1, 58.7, and 116.7 minutes. The speech was separated into utterances, with utterance boundaries being delimited by short pauses of 200 ms or longer. According to this criterion, the training data consisted of 119, 238, 476, 952, and 1,904 utterances respectively. An additional 27.2 minutes (500 utterances) of speech were held out as a test set.

As a measure of the quality of the LM learned by the training process, we used phoneme error rate (PER) when the LM was used to re-score the phoneme lattices of the test set. We chose PER as word-based accuracy may depend heavily on a particular segmentation standard. Given no linguistic information, the PER on the test set was 34.20%. The oracle PER of the phoneme lattice was 8.10%, indicating the lower bound possibly obtainable by LM learning.

Fifty samples of the word sequences $\mathcal{W}$ for each training utterance (and the resulting sufficient statistics S) were taken after 20 iterations of burn-in, the first 10 of which were annealed according to the technique presented by [25]. For the LM scaling factor of Eq. (6), α was set arbitrarily to 5, with values between 5 and 10 producing similar results in preliminary tests.

5.2 Effect of n-gram Context Dependency

In the first experiment, the effect of using context information in the learning process was examined. The n of the HPYLM language model was set to 1, 2, or 3, and n of the HPYLM spelling model was set to 3 for all models. The results with regards to PER are shown in Fig. 7.

First, it can be seen that an LM learned directly from speech was able to improve the accuracy by 7% absolute PER or more compared to a baseline using no linguistic information. This is true even with only 7.9 minutes of training speech. In addition, the results show that the bigram model outperforms the unigram, and the trigram model outperforms the bigram, particularly as the size of the training data increases. We were also able to confirm the observation of [25] that the unigram model tends to undersegment,

[†]Syllable-based decoding was a practical consideration due to the limits of the decoding process, and is not a fundamental part of the proposed method. Phoneme-based decoding will be examined in the future.

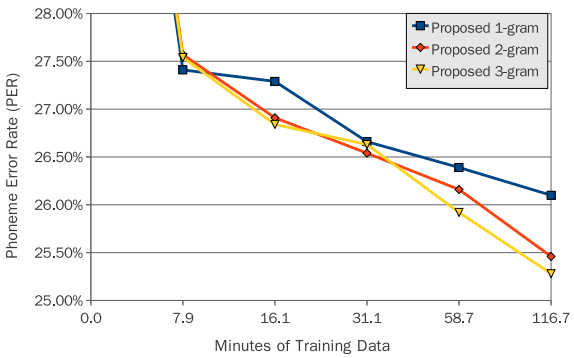


Fig. 7 Phoneme error rate by model order.

Table 1 The size of the vocabulary, and the number of n-grams in the word-based model G, and the phoneme-based model H when trained on 116.7 minutes of speech.

	1-gram	2-gram	3-gram
Vocabulary size	4480	1351	708
G entries	4480	16150	38759
H entries	9624	3869	2426

grouping together "multi-word" phrases instead of actual words. This is reflected in the vocabulary and n-gram sizes of the three models after the final iteration of the learning process, which are displayed in Table 1. It can also be seen that the vocabulary size increases when the LM is given a smaller n, with the lack of complexity in the word-based LM being transferred to the phoneme-based spelling model.

5.3 Effect of Joint and Bayesian Estimation

The proposed method has two major differences from previous methods such as [10], which estimates multigram models from speech lattices. The first is that we are performing joint learning of the lexicon and n-gram context, while multigram models do not consider context, similarly to the 1-gram model presented in this paper [23]. However, it is conceivable that a context insensitive model could be used for learning lexical units, and its results used to build a traditional LM. In order to test the effect of context-sensitive learning, we experiment with not only the proposed 1-gram and 3-gram models from Sect. 5.2, but also use the 1-gram model to acquire samples of $\mathcal{W}$ and use these to train a standard 3-gram LM.

The second major difference is that we are performing learning using Bayesian methods. This allows us to consider the uncertainty of the acquired W through the sum in Eq. (26). Previous multigram approaches are based on maximum likelihood estimation, which only allows for a unique solution to be considered. To test the effect of this, we also take the one-best results acquired by the sampled LMs, but instead of combining them together to create a better result as explained in Sect. 3.3, we simply report the average PER of these one-best results.

Table 2 shows the results of the evaluation (performed on the 116.7 minute training data). It can be seen that

Table 2 The effects on accuracy of the n-gram length used to acquire the lexicon and train the language model, as well as Bayesian sample combination. The proposed method significantly exceeds italicized results according to the two-proportions z-test ($p < 0.05$).

Lexicon	LM	Single	Combined
1-gram	1-gram	*26.28%*	*26.08%*
1-gram	3-gram	*26.06%*	25.41%
3-gram	3-gram	25.85%	**25.28%**

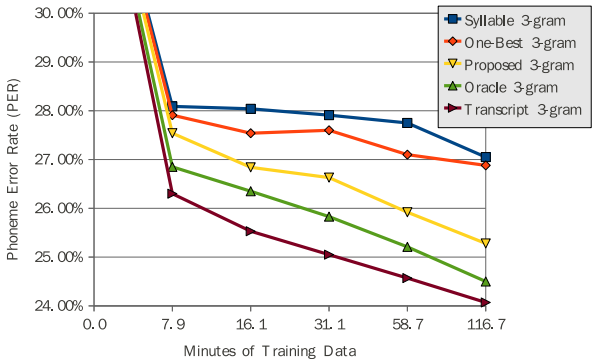


Fig. 8 Phoneme error rate for various training methods.

the proposed method using Bayesian sample combination and incorporating LMs directly into training (3-gram/3-gram/combined) is effective in reducing the error rate compared to a model that does not use these proposed improvements (1-gram/3-gram/single).

5.4 Effect of Lattice Processing

We also compare the proposed lattice processing method with four other LM construction methods. First, we trained a model using the proposed method, but instead of using word lattices, used one-best ASR results to provide a comparison with previous methods that have used one-best results [7], [9]. Second, to examine whether the estimation of word boundaries is necessary when acquiring an LM from speech, we trained a syllable trigram LM using these one-best results. Moreover, we show two other performance results for reference. One is an LM that was built using a human-created verbatim transcription of the utterances. WS and pronunciation annotation were performed with the KyTea toolkit [33], and pronunciations of unknown words were annotated by hand. Trigram language and spelling models were created on the segmented word and phoneme strings using interpolated Kneser-Ney smoothing. For the second reference, we created an "oracle" model by training on the lattice path with the lowest possible PER for each utterance. This demonstrates an upper bound of the accuracy achievable by the proposed model if it picks all the best phoneme sequences in the training lattice.

The PER for the four methods is shown in Fig. 8. It can be seen that the proposed method significantly outperforms the model trained on one-best results, demonstrating that lattice processing is critical in reducing the noise inherent in acoustic matching results. It can also be seen that on one-

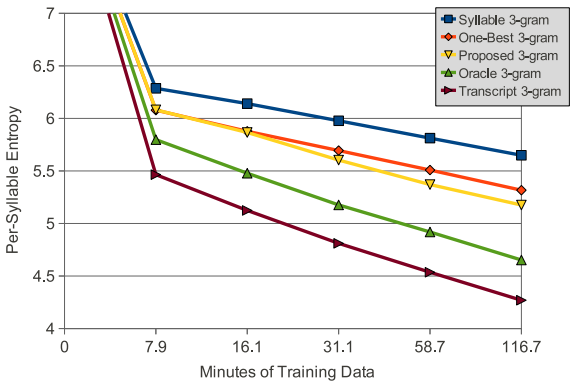


Fig. 9 Entropy comparison for various LM learning methods.

best results, the model using acquired units achieves slightly but consistently better results than the syllable-based LM for all data sizes.

As might be expected, the proposed method does not perform as well as the model trained on gold-standard transcriptions. However, it appears to improve at approximately the same rate as the model trained on the gold-standard transcriptions as more data is added, which is not true for one-best transcriptions. Furthermore, it can be seen that the oracle results fall directly between those achieved by the proposed model and the results on the gold-standard transcriptions. This indicates that approximately one half of the difference between the model learned on continuous speech and that learned from transcripts can be attributed to the lattice error. By expanding the size of the lattice, or directly integrating the calculation of acoustic scores with sampling, it will likely be possible to further close this gap.

Another measure commonly used for evaluating the effectiveness of LMs is cross-entropy on a test set [18]. We show entropy per syllable for the LMs learned with each method in Fig. 9. It can be seen that the proposed method only slightly outperforms the model trained on one-best phoneme recognition results. This difference can be explained by systematic pronunciation variants that are not accounted for in the verbatim transcript. For example, *kangaeteorimasu* ("I am thinking") is often pronounced with a dropped *e* as *kangaetorimasu* in fluent conversation. As a whole word will fail to match the reference, this will have a large effect on entropy results, but less of an effect on PER as only a single phoneme was dropped. In fact, for many applications such as speech analysis or data preparation for acoustic model training, the proposed method, which managed to properly learn pronunciation variants, is preferable to one that matches the transcript correctly.

5.5 Lexical Acquisition Results

Finally, we present a qualitative evaluation of the lexical acquisition results. Typical examples of the words that were acquired in the process of LM learning are shown in Table 3. These are split into four categories: function words, subwords, content words, spoken language expressions.

Table 3 An example of words learned from continuous speech.

Function Words	*no* (genitive marker), *ni* (locative marker), *to* ("and")
Subwords	*ka* (*kyoka* "reinforcement", interrogative marker) *sai* (*kokusai* "international", *seisai* "sanction")
Content Words	*koto* ("thing"), *hanashi* ("speak"), *kangae* ("idea"), *chi-ki* ("region"), *shiteki* ("point out")
Spoken Expressions	*yu-* ("say (colloquial)"), *e-* (filler), *desune* (filler), *mo-shiage* ("say (polite)")

In the resulting vocabulary, function words were the most common of the acquired words, which is reasonable as function words make the majority of the actual spoken utterances. Subwords are the second most frequent category, and generally occur when less frequent content words share a common stem.

An example of the content words discovered by the learning method shows a trend towards the content of discussions made in meetings of the Diet. In particular, *chi-ki* ("region") and *shiteki* ("point out") are good examples of words that are characteristic of Diet speech and acquired by the proposed model. While this result is not surprising, it is significant in that it shows that the proposed method is able to acquire words that match the content of the utterances on which it was trained. In addition to learning the content of the utterances, the proposed model also learned a number of stylistic characteristics of the speech in the form of fillers and colloquial expressions. This is also significant in that these expressions are not included in the official verbatim records in the Diet archives, and thus would not be included in an LM that was simply trained on these texts.

6. Conclusions and Future Work

This paper presented a method for unsupervised learning of an LM given only speech and an acoustic model. Specifically, we adapted a Bayesian model for word segmentation and LM learning so that it could be applied to speech input. This was achieved by formulating all elements of LM learning as WFSTs, which allows for lattices to be used as input to the learning algorithm. We then formulated a Gibbs sampling algorithm that allows for learning over composed lattices that represent acoustic and LM probabilities.

An experimental evaluation showed that LMs acquired from continuous speech with no accompanying transcriptions were able to significantly reduce the error rates of ASR over when no such models were used. We also showed that the proposed technique of joint Bayesian learning of lexical units and an LM over lattices significantly contributes to this improvement.

This work contributes a basic technology that opens up a number of possible directions for future research into practical applications. The first and most immediate application of the proposed method would be for use in semi-supervised learning. In the semi-supervised setting, we have some text already available, but want to discover words from untranscribed speech that may be in new domains, speaking styles, or dialects. This can be formulated in the proposed model

by treating the phoneme sequences Y (and possibly word boundaries W) of existing text as observed variables and the Y and W of untranscribed speech as hidden variables. In addition, if it is possible to create word dictionaries but not a training corpus, these dictionaries could be used as a complement or replacement to the spelling model, allowing the proposed method to favor words that occur in the dictionary.

The combination of the proposed model with information from modalities other than speech is another promising future direction. For example, while the model currently learns words as phoneme strings, it is important to learn the orthographic forms of words for practical use in ASR. One possibility is that speech could be grounded in text data such as television subtitles to learn these orthographic forms. In order to realize this in the proposed model, an additional FST layer that maps between phonetic transcriptions and their orthographic forms could be introduced to allow for a single phonetic word to be mapped into multiple orthographic words and vice-versa.

In addition, the proposed method could be used to discover a lexicon and LM for under-resourced languages with little or no written text. In order to do so, it will be necessary to train not only an LM, but also an acoustic model that is able to recognize the phonemes or tones in the target language. One promising approach is to combine the proposed method with cross-language acoustic model adaptation, an active area of research that allows for acoustic models trained in more resource-rich languages to be adapted to resource-poor languages [31], [34].

The proposed method is also of interest in the framework of computational modeling of lexical acquisition by children. In its current form, which performs multiple passes over the entirety of the data, the proposed model is less cognitively plausible than previous methods that have focused on incremental learning [35]–[37]† However, work by [35] has demonstrated that similar Bayesian methods (which were evaluated on raw text, not acoustic input) can be adapted to an incremental learning framework. This sort of incremental learning algorithm is compatible with the proposed method as well, and may be combined to form a more cognitively plausible model.

The final interesting challenge is how to scale the method to larger data sets. One possible way to improve the efficiency of sampling would be to use beam sampling techniques similar to those developed for non-parametric Markov models [39]. Another promising option is parallel sampling, which would allow sampling to be run on a number of different CPUs simultaneously [40].

†On the other hand, phonemic acquisition is generally considered to occur in the early stages of infancy, prior to lexical acquisition [6], [38], and thus our reliance on a pre-trained acoustic model is largely plausible.

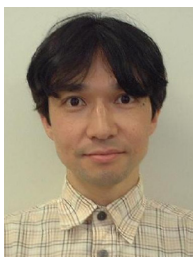
References

[1] D. Tannen, Spoken and Written Language: Exploring Orality and Literacy, ABLEX, 1982.
[2] Y. Akita and T. Kawahara, "Statistical transformation of language and pronunciation models for spontaneous speech recognition," IEEE Trans. Audio Speech Language Process., vol.18, no.6, pp.1539–1549, 2010.
[3] I. Bazzi and J. Glass, "Learning units for domain-independent out-of-vocabulary word modelling," Proc. 7th European Conference on Speech Communication and Technology (EuroSpeech), pp.61–64, 2001.
[4] T. Hirsimaki, M. Creutz, V. Siivola, M. Kurimo, S. Virpioja, and J. Pylkkonen, "Unlimited vocabulary speech recognition with morph language models applied to finnish," Comput. Speech Lang., vol.20, no.4, pp.515–541, 2006.
[5] S. Abney and S. Bird, "The human language project: Building a universal corpus of the world's languages," Proc. 48th Annual Meeting of the Association for Computational Linguistics, pp.88–97, Uppsala, Sweden, July 2010.
[6] D. Roy and A. Pentland, "Learning words from sights and sounds: a computational model," Cognitive Science, vol.26, no.1, pp.113–146, 2002.
[7] C. de Marcken, "The unsupervised acquisition of a lexicon from continuous speech," tech. rep., Massachusetts Institute of Technology, Cambridge, MA, USA, 1995.
[8] S. Deligne and F. Bimbot, "Inference of variable-length linguistic and acoustic units by multigrams," Speech Commun., vol.23, no.3, pp.223–241, 1997.
[9] A. Gorin, D. Petrovska-Delacretaz, G. Riccardi, and J. Wright, "Learning spoken language without transcriptions," Proc. 1999 IEEE Automatic Speech Recognition and Understanding Workshop, 1999.
[10] J. Driesen and H.V. Hamme, "Improving the multigram algorithm by using lattices as input," Proc. 9th Annual Conference of the International Speech Communication Association (InterSpeech), 2008.
[11] L. ten Bosch and B. Cranen, "A computational model for unsupervised word discovery," Proc. 8th Annual Conference of the International Speech Communication Association (InterSpeech), pp.1481–1484, 2007.
[12] A. Park and J. Glass, "Unsupervised pattern discovery in speech," IEEE Trans. Audio Speech Language Process., vol.16, no.1, 2008.
[13] A. Jansen, K. Church, and H. Hermansky, "Towards spoken term discovery at scale with zero resources," Proc. 11th Annual Conference of the International Speech Communication Association (InterSpeech), 2010.
[14] N. Iwahashi, "Language acquisition through a human-robot interface by combining speech, visual, and behavioral information," Inf. Sci., vol.156, no.1-2, pp.109–121, 2003.
[15] C. Yu and D.H. Ballard, "A multimodal learning interface for grounding spoken language in sensory perceptions," ACM Trans. Applied Perception, vol.1, pp.57–80, July 2004.
[16] D. Mochihashi, T. Yamada, and N. Ueda, "Bayesian unsupervised word segmentation with nested Pitman-Yor modeling," Proc. 47th Annual Meeting of the Association for Computational Linguistics, 2009.
[17] Y.W. Teh, "A Bayesian interpretation of interpolated kneser-ney," tech. rep., School of Computing, National Univ. of Singapore, 2006.
[18] J.T. Goodman, "A bit of progress in language modeling," Comput. Speech Lang., vol.15, no.4, pp.403–434, 2001.
[19] S.F. Chen and J. Goodman, "An empirical study of smoothing techniques for language modeling," Proc. 34th Annual Meeting of the Association for Computational Linguistics, 1996.
[20] D.J.C. Mackay and L.C.B. Petoy, "A hierarchical Dirichlet language model," Natural Language Engineering, vol.1, pp.1–19, 1995.
[21] J. Pitman and M. Yor, "The two-parameter Poisson-Dirichlet distri-

bution derived from a stable subordinator," The Annals of Probability, vol.25, no.2, pp.855–900, 1997.
[22] M. Mohri, F. Pereira, and M. Riley, "Speech recognition with weighted finite-state transducers," in Handbook on speech processing and speech communication, Part E: Speech recognition, 2008.
[23] F. Bimbot, R. Pieraccini, E. Levin, and B. Atal, "Variable-length sequence modeling: Multigrams," IEEE Signal Process. Lett., vol.2, no.6, pp.111–113, 1995.
[24] M.R. Brent, "An efficient, probabilistically sound algorithm for segmentation and word discovery," Mach. Learn., vol.34, pp.71–105, 1999.
[25] S. Goldwater, T.L. Griffiths, and M. Johnson, "A Bayesian framework for word segmentation: Exploring the effects of context," Cognition, vol.112, no.1, pp.21–54, 2009.
[26] H. Poon, C. Cherry, and K. Toutanova, "Unsupervised morphological segmentation with log-linear models," Proc. North American Chapter of the Association for Computational Linguistics - Human Language Technology (NAACL HLT), pp.209–217, 2009.
[27] S. Geman and D. Geman, "Stochastic relaxation, Gibbs distributions and the Bayesian restoration of images," IEEE Trans. Pattern Anal. Mach. Intell., vol.6, no.6, pp.721–741, 1984.
[28] D.J. MacKay, Information theory, inference, and learning algorithms, pp.357–386, Cambridge University Press, 2003.
[29] C.S. Jensen, U. Kjærulff, and A. Kong, "Blocking Gibbs sampling in very large probabilistic expert systems," Int. J. Human Comput. Studies, vol.42, no.6, pp.647–666, 1995.
[30] J. Fiscus, "A post-processing system to yield reduced word error rates: Recognizer output voting error reduction (ROVER)," Proc. 1997 IEEE Automatic Speech Recognition and Understanding Workshop, 1997.
[31] L. Lamel, J. Gauvain, and G. Adda, "Lightly supervised and unsupervised acoustic model training," Comput. Speech Lang., vol.16, pp.115–129, 2002.
[32] J.R. Glass, Finding acoustic regularities in speech: Application to phonetic recognition, Ph.D. thesis, Massachusetts Institute of Technology, Cambridge, MA, USA, 1988.
[33] G. Neubig and S. Mori, "Word-based partial annotation for efficient corpus construction," Proc. 7th International Conference on Language Resources and Evaluation, 2010.
[34] T. Schultz and A. Waibel, "Language-independent and language-adaptive acoustic modeling for speech recognition," Speech Commun., vol.35, no.1, pp.31–52, 2001.
[35] L. Pearl, S. Goldwater, and M. Steyvers, "How ideal are we? incorporating human limitations into Bayesian models of word segmentation," Proc. 34th Annual Boston University Conference on Child Language Development, pp.315–326, Somerville, MA, 2010.
[36] F.R. McInnes and S. Goldwater, "Unsupervised extraction of recurring words from infant-directed speech," Proc. 33rd Annual Conference of the Cognitive Science Society, 2011.
[37] O. Räsänen, "A computational model of word segmentation from continuous speech using transitional probabilities of atomic acoustic events," Cognition, vol.120, no.2, pp.149–176, 2011.
[38] P.D. Eimas, E.R. Siqueland, P. Jusczyk, and J. Vigorito, "Speech perception in infants," Science, vol.171, no.3968, p.303, 1971.
[39] J. Van Gael, Y. Saatci, Y. Teh, and Z. Ghahramani, "Beam sampling for the infinite hidden Markov model," Proc. 25th International Conference on Machine Learning, 2008.
[40] A. Asuncion, P. Smyth, and M. Welling, "Asynchronous distributed learning of topic models," Proc. 22nd Annual Conference on Neural Information Processing Systems, vol.21, 2008.

Graham Neubig received his B.E. from University of Illinois, Urbana-Champaign, U.S.A, in 2005, and his M.E. in informatics from Kyoto University, Kyoto, Japan in 2010, where he is currently pursuing his Ph.D. He is a recipient of the JSPS Research Fellowship for Young Scientists (DC1). His research interests include speech and natural language processing, with a focus on unsupervised learning for applications such as automatic speech recognition and machine translation.

Masato Mimura received the B.E. and M.E. degrees from Kyoto University, Kyoto, Japan, in 1996 and 2000, respectively. Currently, he is a researcher in the Academic Center for Computing and Media Studies, Kyoto University. His research interests include spontaneous speech recognition and spoken language processing.



Shinsuke Mori received B.S., M.S., and Ph.D. degrees in electrical engineering from Kyoto University, Kyoto, Japan in 1993, 1995, and 1998, respectively. After joining Tokyo Research Laboratory of International Business Machines (IBM) in 1998, he studied the language model and its application to speech recognition and language processing. He is currently an associate professor of Academic Center for Computing and Media Studies, Kyoto University.

Tatsuya Kawahara received B.E. in 1987, M.E. in 1989, and Ph.D. in 1995, all in information science, from Kyoto University, Kyoto, Japan. In 1990, he became a Research Associate in the Department of Information Science, Kyoto University. From 1995 to 1996, he was a Visiting Researcher at Bell Laboratories, Murray Hill, NJ, USA. Currently, he is a Professor in the Academic Center for Computing and Media Studies and an Affiliated Professor in the School of Informatics, Kyoto University. He has also been an Invited Researcher at ATR and NICT. He has published more than 200 technical papers on speech recognition, spoken language processing, and spoken dialogue systems. He has been managing several speech-related projects in Japan including a free large vocabulary continuous speech recognition software project (http://julius.sourceforge.jp/). Dr. Kawahara received the 1997 Awaya Memorial Award from the Acoustical Society of Japan and the 2000 Sakai Memorial Award from the Information Processing Society of Japan. From 2003 to 2006, he was a member of IEEE SPS Speech Technical Committee. From 2011, he is a secretary of IEEE SPS Japan Chapter. He was a general chair of IEEE Automatic Speech Recognition & Understanding workshop (ASRU 2007). He also served as a tutorial chair of INTERSPEECH 2010. He is a senior member of IEEE.