Mixture of Latent Words Language Models for Domain Adaptation

Ryo Masumura, Taichi Asami, Takanobu Oba, Hirokazu Masataki, Sumitaka Sakauchi

NTT Media Intelligence Laboratories, NTT Corporation, Japan
{masumura.ryo, asami.taichi, oba.takanobu, masataki.hirokazu, sakauchi.sumitaka}@lab.ntt.co.jp

Abstract

This paper introduces a novel language model (LM) adaptation method based on mixture of latent word language models (LWLMs). LMs are often constructed as mixture of n-gram models, whose mixture weights are optimized using target domain data. However, n-gram mixture modeling is not flexible enough for domain adaptation because model merger is conducted on the observed word space. Since the words in out-of-domain LMs often differ from those in the target domain LM, it is hard for out-of-domain LMs to offer adequate adaptation performance. Our solution is to carry out model merger in a latent variable space. The latent variables in the LWLMs are represented as specific words selected from the observed word space, so LWLMs can share a common latent variable space and we can realize mixture modeling with consideration of the latent variable space. Following this change, this paper also describes a method to estimate mixture weights for LWLM mixture modeling. We use a sampling technique based on the Bayesian criterion in place of the conventional expectation maximization algorithm. Our experiments show that the LWLM mixture modeling is more effective than n-gram mixture modeling.

Index Terms: language model adaptation, n-gram mixture modeling, latent words language models, latent variable space

1. Introduction

With the recent progress of speech recognition technologies, further extension of their application field can be expected. To this end, technology that can build effective language models (LMs) using limited target domain data is needed since large amounts of specific domain data are not often available. There are two approaches to realize this technology. One is robust modeling; it can realize high performance even when the training data is restricted [1]. The other is domain adaptation, which specializes in the target domain using out-of-domain data [2].

For the robust modeling, various methods have been studied. In automatic speech recognition (ASR), word n-gram LMs that directly model the observed word space are often used. Even when there is little training data, n-gram LMs can robustly perform probability estimation based on model smoothing [3]. In addition, further robustness can be achieved by reducing the dimensionality of the modeling space [4, 5, 6, 7, 8]. We pay particular attention to latent words language models (LWLMs) which have latent variables called latent words [9]. LWLM has a soft clustering structure and a vast latent variable space whose size is equivalent to the observed word space. We previously applied LWLM to ASR and confirmed that LWLM outperforms n-gram LM in multiple domain [10, 11]. Our work found that consideration of the latent variable space is important for robust modeling.

This paper, on the other hand, focuses on domain adaptation. One of the most popular frameworks in LM adaptation is mixture modeling [12, 13, 14]. In this framework, target-domain LM and LMs, which are individually trained from out-of-domain data, are combined with mixture weighting [15]. Especially, since n-gram mixture models can be expressed as a single back-off n-gram model, the n-gram mixture modeling is used in various cases [16].

To improve mixture modeling, a lot research has focused on determining the mixture weights. A fundamental technique is maximum likelihood (ML) criterion [17]. If we obtain a sufficient amount of target domain data (development data), we can use the expectation maximization (EM) algorithm to estimate mixture weights. Another approach is based on minimum Bayes risk criterion; it can be generalize to yield a variety of error cost functions [18]. Moreover, context-dependent weighting schemes have been proposed to secure further improvements [19, 20]. These frameworks aim to advancing mixture modeling in the observed word space.

However, mixtures in the observed word space do not support flexible adaptation. In simple n-gram mixture modeling, each n-gram model is combined in an observation word space since a word is directly in agreement with a state. The words in out-of-domain LMs often differ from those in the target domain LM, so it is unlikely that a state can be shared by model mixtures. Thus, adaptation is hard to secure with out-of-domain LMs, and generalization performance will fall seriously because the adapted model will match the objective task too much.

Our solution is to realize a method in which model merging is conducted in a latent variable space in common with the robust modeling. Since a word is mapped into a latent variable space, we can expect to realize state sharing between models more flexibly than is possible in the observed word space. To this end, this paper introduces the LWLM mixture modeling. The latent variables in usual class-based n-gram LMs are only model-dependent indices, so each model has a different latent variable space [5, 6, 7, 8]. Therefore, conventional class-based n-gram mixture modeling have to be performed in the observed word space [21, 22, 23]. On the other hand, latent variables in LWLMs are represented as specific latent word, multiple LWLMs can share the common latent variable space.

For merging the LWLMs in the latent variable space, we also need a different kind of method to define mixture weights. In n-gram mixture modeling, the ML criterion can be used because generation probabilities of each word of development data can be directly calculated. Unfortunately, this advantage is offset by the fact the latent word sequence of development data cannot be determined uniquely. To estimate optimal mixture weights, this paper proposes a framework based on the Bayesian criterion. The Bayesian criterion can be flexibly applied to various model structures and allows us to use sampling techniques. We use Gibbs sampling and estimate mixture weights by sam-
pling the latent word sequence and model index sequence underly-
ing the development data [24].

This paper is organized as follows. First, n-gram mixture
modeling and its optimization framework based on ML criterion
are briefly described in Section 2. Section 3 denotes LWLM and
its ASR implementation. Section 4 explains the LWLM mixture
modeling and its optimization method. Section 5 describes our
experiments and Section 6 concludes this paper.

2. Domain adaptation based on mixture of n-gram LMs

2.1. N-gram mixture modeling

N-gram LMs defines probability distribution \( P(w_k|u_k, \theta) \) over
current word \( w_k \) given context \( u_k = u_{k-n+1}, \ldots, u_{k-1} \)
and its model parameter \( \theta \). For instance, hierarchical Pitman-
Yor LM is known to be typical n-gram LM [25, 26]. An n-gram mix-
ture model is constructed by combining several n-gram models
trained using different sources. A graphic representation of an
n-gram mixture model is shown in Figure 1; the model index
is represented as a single back-off n-gram model [16].

As shown in Figure 2, the latent variable, called latent word \( h_k \),
is generated from a transition probability distribution given context
\( l_k = h_{k-n+1}, \ldots, h_{k-1} \). Then, observed word \( w_k \) is generated from an emission probability
distribution given the latent word \( h_k \).

LWLM has a soft clustering structure that differs from a
simple hard clustering structure. In LWLM, each word belongs
to all classes. In addition, a latent word is expressed as a specific
word that can be selected from the observed word space. Thus,
the size of observed word space is equal to the latent variable
space.

Bayesian inference of LWLM produces the predictive dis-
bution of observed word sequence \( w \). In this paper, the pre-
dictive probability distribution is approximately calculated by a
point estimation:

\[
P(w|\theta) = \sum_h \int P(h|\theta) P(w|h, \theta) P(h) d\theta,
\]

\[
= \sum_h \int P(\theta) \prod_{k=1}^{K} P(w_k|h_k, \theta) P(h_k|l_k, \theta) d\theta,
\]

\[
\approx \sum_{h} \prod_{k=1}^{K} P(w_k|h_k, \theta) P(h_k|l_k, \theta),
\]

where \( \theta \) is a model parameter of LWLM, \( P(h_k|l_k, \theta) \) is the n-
gram model for latent words, and \( P(w_k|h_k, \theta) \) models the
dependency between the observed word and the latent word.
\( h \) is a latent word sequence. We use the hierarchical Pitman-
Yor prior for the transition probability distribution and Dirich-
let prior for the emission probability distribution [10, 11]. For
training LWLM, we have to infer a latent word sequence \( h \)
behind training data \( w \). To this end, Gibbs sampling can be
used.

To implement LWLMs to ASR, we approximate them to
yield a structure suitable for ASR [10]. The approximate model
is constructed by randomly generating text data according to a
stochastic process and training a standard word based n-gram
LM.

4. Domain adaptation based on mixture of LWLMs

4.1. LWLM mixture modeling

In this paper, we propose a LWLM mixture modeling. A
graphic rendering of LWLM mixture models is shown in Figure
3. As shown, LWLM mixture modeling can be considered to be
the union of Figure 1 and Figure 2, and the relation between
LWLMs occurs in the latent variable space. In the generative
process of LWLM mixture models, a model index that corre-
sponds to each LWLM index is first generated. Then, latent

Figure 1: Graphical representation of n-gram mixture models.

Figure 2: Graphical representation of LWLMs.
and observed words are generated based on the stochastic process of the selected LWLM. Briefly, latent word sequence and observed word sequence are generated dependent on the model index sequence.

In LWLM mixture models, the predictive distribution of observed words, w, is defined as:

$$P(w|z) = \sum_z P(w|h, z, \Theta) P(h|z, \Theta) P(z|\Theta),$$

$$= \prod_{k=1}^{K} \sum_{z_k} P(w_k|h_k, \theta_{z_k}) P(h_k|l_k, \theta_{z_k}) P(z_k|\phi),$$

where $\Theta$ is the model parameter of the LWLM mixture model. In this equation, $P(z|\phi)$ can be estimated from the development set. This equation is based on the characteristics that LWLMs share a common latent variable space.

### 4.2. Adaptation on latent variable space

To optimize $P(z|\phi)$ using the development set, we cannot employ the ML criterion because the latent word sequence is an underspecified variable. If we use the ML criterion, we have to consider all possible latent word assignments since LWLM has a soft clustering structure. It is computationally and analytically intractable to calculate the expectation value. Therefore, we use the Bayesian criterion and a sampling based procedure that is compatible with LWLM training.

In Bayesian criterion, Dirichlet prior distribution is put on $P(z|\phi)$. In this case, we can estimate $P(z|\phi)$ by using Gibbs sampling to assign latent word sequence $h$ and model index sequence $z$ to development set $w$. The conditional probability of possible values for latent word $h_k$ is given by:

$$P(h_k|w, h^{-k}, z)$$

$$\sim P(w_k|h_k, \theta_{z_k}) \prod_{j=k}^{k+n-1} P(h_j|l_j, \theta_{z_j}),$$

where $h^{-k}$ represents all latent words except for $h_k$. In a similar way, the conditional probability of possible values for model index $z_k$ is obtained as:

$$P(z_k|w, h, z^{-k})$$

$$\sim P(w_k|h_k, \theta_{z_k}) P(h_k|l_k, \theta_{z_k}) P(z_k|z^{-k}),$$

where $z^{-k}$ represents model index sequence except for $z_k$. Gibbs sampling can be used to sample new values for the model index and the latent variable according to these two distributions and place them at position $k$.

Once model index sequence is concluded, $P(z_k|\phi)$ is calculated as:

$$P(z_k|\phi) = \frac{c(z_k; z) + \beta}{\sum_{z'} c(z'; z) + \beta},$$

where $c(z_k; z)$ means the number of model index $z_k$ in $z$. $\beta$ is hyper parameter for Dirichlet distribution.

In Bayesian criterion, optimized value $P(z_k|\phi)$ is estimated by Monte Carlo integration. Multiple model index sequences sampled after the burn-in period are defined as $z^1, \ldots, z^S$. $P(z_k|\phi)$ is estimated as:

$$P(z_k|\phi) = \frac{1}{S} \sum_{s=1}^{S} P(z_k|z^s).$$

If $\beta$ approaches 0, the Bayesian criterion is equivalent to the ML criterion.

### 4.3. Approximation 0 for ASR

To apply an LWLM mixture model to ASR, a special technique is needed as well as a single LWLM. We also approximate the LWLM mixture model to create a structure suitable for ASR as in [10]. The approximate model is constructed by randomly generating text data according to a stochastic process and training a standard back-off n-gram model. Each $\kappa$-th variable is generated as:

$$z_\kappa \sim P(z_\kappa|\phi),$$

$$h_\kappa \sim P(h_\kappa|l_\kappa, \theta_{z_\kappa}),$$

$$w_\kappa \sim P(w_\kappa|h_\kappa, \theta_{z_\kappa}).$$

By iterating the above $K$ times, we can generate the model index sequence, latent word sequence and observed word sequence. The observed word sequence is used for back-off n-gram model estimation.

### 5. Experiments

#### 5.1. Experimental conditions

Our experiments employed the Corpus of Spontaneous Japanese (CSI) [27]. We divided the training and test data into two kinds, academic lectures and extemporaneous lectures. Target domain was set to the former. Details of the experimental data set are shown in Table 1.

<table>
<thead>
<tr>
<th>Data</th>
<th>Kinds</th>
<th># of words</th>
</tr>
</thead>
<tbody>
<tr>
<td>Training (target domain)</td>
<td>Academic</td>
<td>3,468,133</td>
</tr>
<tr>
<td>Training (out-of domain)</td>
<td>Extemporaneous</td>
<td>3,847,816</td>
</tr>
<tr>
<td>Development</td>
<td>Academic</td>
<td>28,046</td>
</tr>
<tr>
<td>Test (target domain)</td>
<td>Academic</td>
<td>27,907</td>
</tr>
<tr>
<td>Test (out-of domain)</td>
<td>Extemporaneous</td>
<td>18,251</td>
</tr>
</tbody>
</table>

We used an acoustic model based on hidden Markov models with deep neural networks (DNN-HMM) [28]. The trained DNN-HMM had 7 hidden layers of 2048 nodes and 3874 outputs. The speech recognition decoder was VoiceRex, a WFST-based decoder [29, 30]. JTAG was used as the morpheme analyzer to split sentences into words [31].

We evaluated both robust modeling and domain adaptation. We compared three methods as the robust modeling which only uses the target domain training data.
### Table 2: Experimental results.

<table>
<thead>
<tr>
<th></th>
<th>Development (target domain)</th>
<th>Test (target domain)</th>
<th>Test (out-of-domain)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>PPL</td>
<td>WER %</td>
<td>PPL</td>
</tr>
<tr>
<td>1. HPYLM</td>
<td>70.57</td>
<td>20.80</td>
<td>62.85</td>
</tr>
<tr>
<td>2. Approximated LWLM</td>
<td>70.02</td>
<td>20.72</td>
<td>62.34</td>
</tr>
<tr>
<td>3. HPYLM + Approximated LWLM</td>
<td><strong>65.30</strong></td>
<td><strong>19.27</strong></td>
<td><strong>58.25</strong></td>
</tr>
<tr>
<td>4. Mixture of HPYLMs</td>
<td>71.68</td>
<td>18.78</td>
<td>64.19</td>
</tr>
<tr>
<td>5. Mixture of approximated LWLMs</td>
<td>72.83</td>
<td>18.56</td>
<td>64.57</td>
</tr>
<tr>
<td>6. Mixture of LWLMs</td>
<td>72.72</td>
<td>18.45</td>
<td>64.39</td>
</tr>
<tr>
<td>7. Mixture of HPYLMs + Mixture of approximated LWLMs</td>
<td>67.52</td>
<td>17.88</td>
<td>60.45</td>
</tr>
<tr>
<td>8. Mixture of HPYLMs + Mixture of LWLMs</td>
<td><strong>67.38</strong></td>
<td><strong>17.64</strong></td>
<td><strong>60.19</strong></td>
</tr>
</tbody>
</table>

1. **HPYLM**: Hierarchical Pitman-Yor LM constructed from the target domain training data [26].
2. **Approximated LWLM**: Approximated LWLM constructed from the target domain training data, we generated one Giga words and approximated as a hierarchical Pitman-Yor LM [10].
3. **HPYLM + Approximated LWLM**: Mixed model which combined both HPYLM and approximated LWLM [10].

Each model was a trigram model and count cutoff pruning was not used, vocabulary size was 40,725. We used 200 iterations for burn-in, and collected 10 samples to train HPYLM. We also used 500 iterations for burn-in, and collected 10 samples to train LWLM. Besides, we compared four methods as the domain adaptation, which has respectively independent mixture weights in the latent variable space. We also proposed a method to estimate mixture weights for LWLM mixture modeling. In this paper, we proposed LWLM mixture modeling for language model adaptation. Latent variables in LWLMs are represented as specific words that can be selected from the observed word space, so we can realize mixture modeling with consideration of the latent variable space. We also proposed a method to estimate mixture weights for LWLM mixture modeling. This framework can realize optimization for development sets as well as n-gram mixture modeling. Experiments on the Corpus of Spontaneous Japanese showed that LWLM mixture modeling achieves higher performance than n-gram mixture modeling. This result shows that mixture modeling in the latent variable space can realize more flexible adaptation than that in the observed word space. Actually, in mixture modeling on a latent variable space, the mixture weight for out-of-domain LM is comparatively high compared with that in an observed word space. (The mixture weight for out-of-domain LM is 0.09. That in LWLM mixture modeling was 0.13.) Although we also evaluated the case where mixture of LWLMs and mixture of HPYLMs were combined, the improvement was not acquired compared to the mixture of LWLMs. It turned out that the proposed LWLM mixture modeling can achieve improvement for all the target domain and the out-of-domain compared with the n-gram mixture modeling.

### 6. Conclusions

In this paper, we proposed LWLM mixture modeling for language model adaptation. Latent variables in LWLMs are represented as specific words that can be selected from the observed word space, so we can realize mixture modeling with consideration of the latent variable space. We also proposed a method to estimate mixture weights for LWLM mixture modeling. This framework can realize optimization for development sets as well as n-gram mixture modeling. Experiments on the Corpus of Spontaneous Japanese showed that LWLM mixture modeling achieves higher performance than n-gram mixture modeling. Furthermore, the proposed domain adaptation was robust with regard to out-of-domain data.

For future work, we plan to expand LWLM mixture modeling, which has respectively independent mixture weights in the transition probability distribution and the emission probability distribution for further robust domain adaptation.
7. References


