Structural Bayesian Language Modeling and Adaptation

Sibel Yaman¹, Jen-Tzung Chien², Chin-Hui Lee³

¹Department of Electrical and Computer Engineering, Georgia Institute of Technology, GA, USA
²Department of Computer Science and Information Engineering, National Cheng Kung University, Tainan, Taiwan

Abstract

We propose a language modeling and adaptation framework using Bayesian structural maximum a posteriori (SMAP) principle, in which each n-gram event is embedded in a branch of a tree structure. The nodes in the first layer of this tree structure represent the unigrams, and those in the second layer represent the bigrams, and so on. Each node in the tree structure has an associated hyper-parameter representing the information about the prior distribution, and a count representing the number of times the word sequence occurs in the domain-specific data. In general, the hyper-parameters depend on the observation frequency of not only the node event but also its parent node of lower order n-gram event. Our automatic speech recognition experiments using the Wall Street Journal corpus verify that the proposed SMAP language model adaptation achieves a 5.6% relative improvement over maximum likelihood language models obtained with the same training and adaptation data sets.

Index Terms: language model adaptation, Bayesian language modeling, Bayesian learning, automatic speech recognition and understanding

1. Introduction

Statistical n-grams are the state-of-art language models (LMs) for automatic speech recognition (ASR) systems. LMs capable of characterizing the domain-specific knowledge are critical components of ASR systems to achieve desirable word errors rates (WERs). It is a widely accepted consensus that high-order n-gram LMs yield better ASR performance. However, high-order n-gram LMs often suffer from the sparsity of training data, and from the lack of domain knowledge. It is for these reasons that trigram LMs are frequently used, and n-grams of order 4 and more are seldom considered.

When there is only a limited amount of domain-specific data, linear interpolation of two models, one trained on the general-domain data and the other trained on the domain-specific data, produces reliable LMs. MAP-based LM adaptation has recently been argued to be a more effective way to make best use of the valuable domain-specific data [1]. As an alternative, structural LMs take into account the hierarchical nature of natural language by leveraging syntactic knowledge. In [2], it was found that using a well-trained structured LM as a background LM and re-estimating the parameters using an adaptation set produced better results than training a model from scratch using the adaptation set. In [3], the application of binary decision trees to language modeling is investigated, and it is concluded that the tree-based LM provided a lower perplexity and fewer bad probabilities than an equivalent trigram model.

In [4], a structural maximum a posteriori (SMAP) framework is described for speaker adaptation, where the probability density functions for model parameters at one level are used as priors for those of the parameters at adjacent levels. In this paper, we propose a similar SMAP framework for LM adaptation where the n-grams are assembled in a tree structure. In accordance with the proposed SMAP LM framework, each n-gram corresponds to a branch in a tree, and the n-gram words are seen as the nodes through the tree branch. We evaluate the goodness of the obtained LMs in the ASR experiments using the Wall Street Journal (WSJ) corpus. Since no adaptation data is provided with the WSJ corpus, we artificially extract an adaptation set from the available LM training data. Our experiments indicate that the proposed framework is able to reduce WER from 5.01% to 4.73%, which corresponds to 5.6% relative improvement over maximum likelihood language modeling.

2. SMAP Language Modeling and Adaptation

In automatic speech recognition systems, the best word sequence \( \hat{W} \) corresponding to an acoustic signal, \( X \), is found by the plug-in MAP decoding algorithm

\[
\hat{W} = \arg \max_{W} P(W|X) = \arg \max_{W} P(X|W)P(W). \tag{1}
\]

The LM component, \( P(W) \), is used to perform sentence-level matching in MAP decoding. Statistical n-gram LMs are currently the state-of-the-art, with \( P(W) \) computed as

\[
P(W) = \prod_{i=1}^{K} P(\omega_i|h_{\omega_i} = \omega_{i-n+1}^{i-1}), \tag{2}
\]

where \( h_{\omega_i} \) denotes the history of the word \( \omega_i \), i.e. its preceding words, and \( \omega_{j}^{i} \) denotes the sequence of words \( \omega_j \) through \( \omega_i \). Statistical n-grams basically suffer from three kinds of insufficiencies, namely the lack of sufficient training data, the lack of domain knowledge, and long-distance dependency. The problem with insufficient domain knowledge can be overcome by LM adaptation.

2.1. Hierarchical Language Models

In the proposed framework, the n-grams are assembled in tree structures where the nodes in the first layer represent the unigrams, and those in the second layer represent the bigrams, and so on. The goal in building hierarchical models is to adaptively estimate the posterior probabilities of n-grams in a top-down manner.
Consider Figure 1 for estimating the probability of the word $\omega_i$ given its history, i.e., words $\omega_1$ through $\omega_i$. The word at the root node, $\omega_1$, is the history node of word $\omega_i$ at the first layer. The words $\omega_2$ and $\omega_3$ together with their own history nodes are the history nodes of $\omega_i$ at the second and third layer, respectively. The words $\omega_4$ and $\omega_5$ are embedded in the tree structure where $\omega_4$ is followed by $\omega_5$ and $\omega_3$ is followed by $\omega_4$.

It is this tree structure that make it possible to merge the information conveyed by the general-domain data and the information conveyed by domain-specific data. As will be developed in the following sections, the general-domain data is first arranged in hierarchical manner, and is used to find prior probability distribution of the $n$-grams. The information extracted from domain-specific data is carefully inserted into this hierarchical structure to find MAP probabilities of $n$-grams.

2.2. LM Probability Estimation in a Tree Node

Consider a branch $\beta$ in a tree represented by the $(n-1)$-gram $h^{(n-1)}_{\omega_i}$ followed by word $\omega_i$ at the $\ell$th layer. Let $c(h^{(n-1)}_{\omega_i})$ denote the number of times the $(n-1)$-gram $h^{(n-1)}_{\omega_i}$ is observed, and $c(h^{(n-1)}_{\omega_i}, \omega_i)$ denote the number of times $h^{(n-1)}_{\omega_i}$ is followed by $\omega_i$. Also, let $\phi_{h^{(n-1)}_{\omega_i}}$ denote the hyper-parameter representing the prior information associated with the node at the $\ell$th layer in the branch, $\beta$, and $\theta_{h^{(n-1)}_{\omega_i}}$ denote the probability of $\omega_i$ at the $\ell$th layer given its history, i.e.,

$$\theta_{h^{(n-1)}_{\omega_i}} = P(\omega_i| h^{(n-1)}_{\omega_i} = \omega_1, \ldots, \omega_{\ell-1}).$$

When an appropriate prior distribution is known, the probabilities $\theta_{h^{(n-1)}_{\omega_i}}$ can be found by maximizing the a posteriori probability given the observations $W$. This corresponds to solving the problem

$$\theta_{h^{(n-1)}_{\omega_i}} = \arg \max_\theta P(\theta | W) = \arg \max_\theta P(W | \theta) g(\theta).$$

The distribution with probabilities of discrete words is analogous to the distribution with mixture components within a mixture density. Hence, MAP language modeling and adaptation problem is very similar to the MAP estimation of the mixture weights of a mixture distribution. For the case of $n$-grams, the underlying distribution is the multinomial density, for which the conjugate prior density is the Dirichlet density. Following the motivation and derivation in [5], the prior density is given by

$$g(\theta_{h^{(n-1)}_{\omega_i}}) = g(P(\omega_i| h^{(n-1)}_{\omega_i})) = g(\phi_{h^{(n-1)}_{\omega_i}}) \propto \theta_{h^{(n-1)}_{\omega_i}}^{-1}. $$

The hyper-parameters for the root nodes, i.e. for $\ell = 1$ are computed as

$$\phi_{h^{(1)}_{\omega_i}} = 1 + c(h^{(1)}_{\omega_i}),$$

and the hyper-parameters are propagated in a top-down manner, yielding for $\ell = 2, 3, \ldots$,

$$\phi_{h^{(\ell)}_{\omega_i}}^{(\ell-1)} = c(h^{(\ell-1)}_{\omega_i}, \omega_i) + \gamma[\phi_{h^{(\ell-1)}_{\omega_i}}^{(\ell-1)} - 1] + 1. $$

A non-negative weight, $\rho$, controlling the effect of the prior density is incorporated into this process. The greater the weight $\rho$ is, the more the prior information is depended upon. When $\rho = 0$, the prior information is discarded, and when $\rho \to \infty$, only the prior information is used. With the weight factor included, the MAP estimation problem is

$$\theta_{h^{(\ell)}_{\omega_i}} = \arg \max_\theta \log P(W | \theta) + \rho \log g(\theta),$$

subject to $\sum_{\omega_i \in \Omega_{h^{(\ell)}_{\omega_i}}} \theta_{h^{(\ell)}_{\omega_i}} = 1$.

The problem in Eq. (10) is to be solved for all tree layers, $\ell = 1, 2, \ldots, n$, for each individual tree separately. By incorporating a Lagrangian factor $\lambda$ for the constraint, the problem in Eq. (10) has a closed-form solution given by

$$\theta_{h^{(\ell)}_{\omega_i}} = \frac{c(h^{(\ell)}_{\omega_i}, \omega_i) + \rho(\phi^{(\ell)}_{h^{(\ell)}_{\omega_i}} - 1)}{c(h^{(\ell)}_{h^{(\ell)}_{\omega_i}}) + \sum_{\omega_i \in \Omega_{h^{(\ell)}_{\omega_i}}} [\rho(\phi^{(\ell)}_{h^{(\ell)}_{\omega_i}} - 1)]}$$

In case there is no domain-specific knowledge, the LM estimates predicted by Eq. (11) are to be used to find MAP probabilities in an $n$-gram model. When LM adaptation data is made available, the notation in the above formulation is slightly changed to reflect the fact that the training data is used to estimate the hyper-parameters of the prior distribution, and adaptation data sentences $W$ replace the training data sentences $\tilde{W}$. The replacement of the $W$ by $\tilde{W}$ means that the counts $c(\cdot)$ obtained from the training data are to be changed with counts $\tilde{c}(\cdot)$ obtained from the adaptation data in Eqs. (10) and (11).

3. Selection of Adaptation Set

The Wall Street Journal (WSJ10) corpus was designed to provide a wealth of general-purpose speech data with large vocabularies [6]. It has a set of over 1.6 million standardized sentences for LM training. The WSJ10 corpus is scalable, and built to accommodate variable size large vocabularies (5K, 20K and 60K words). In this paper, all results are reported using the 5K non-verbalized punctuation closed vocabulary. The availability of large amounts of text material enabled having statistical benchmark LMs to be generated. We found the WER using the benchmark trigram LM to be 5.01% on the November 92 test set consisting of 330 utterances.

There is no LM adaptation set provided within the WSJ corpus. Therefore, the rest of this section is devoted to the construction of an appropriate artificial adaptation set to illustrate
the proposed SMAP LM adaptation framework. Preliminary results are listed in the following to illustrate the selection effectiveness of the adaptation set. Detailed experimental results will be reported in Section 4.

### 3.1. An Irrelevant Adaptation Set

As a first attempt to find an adaptation set, we randomly selected 1 out-of-every 10 sentences from the original training set, $\mathcal{O}$. The remaining sentences, which constitute 90% of $\mathcal{O}$, are used as the reduced training data, $\mathcal{S}_i$. As it will turn out, the adaptation set, $\mathcal{O} \setminus \mathcal{S}_i$, constructed in this way was non-informative about the test domain. Thus, we call this adaptation set an irrelevant adaptation set, and denote it as $\mathcal{A}_i$. Despite its weakness to provide useful knowledge about the test domain, experiencing with it led us construct a relevant adaptation set.

In similar speaker adaptation task, the performance improves as more and more speaker data is obtained. To evaluate the effect of more adaptation data on the performance of LM adaptation, we constructed five subsets of the adaptation set such that $\mathcal{S}_i \subset \mathcal{S}_i' \subset \mathcal{S}_i'' \subset \cdots \subset \mathcal{A}_i$.

The advantage we expect from SMAP language adaptation is being able to reduce the WER to a level that is lower than what could be achieved with ML modeling. As summarized in Table 3 of Section 4, the ML model trained on $\mathcal{S}_i$, which covers 90% of $\mathcal{O}$, yields 5.04% WER. When we train an ML model on $\mathcal{O}$, the performance improves to just 5.01% WER. The experimental result of using $\mathcal{A}_i$ for LM adaptation is shown in Figure 2 of Section 4. Consequently, the WER stays roughly the same regardless of whether $\mathcal{A}_i$ is used for adaptation or as additional data for ML modeling.

These experimental results indicated that the constructed adaptation set, $\mathcal{A}_i$, has similar characteristics to the training set, and hence, was non-informative about the test domain. Concerning with the best WER that could be achieved, we used the test data in adaptation, and found that a WER of 1.73% is achievable if we had exact knowledge about the test domain. In this way, we conclude that $\mathcal{A}_i$ provides additional informative test domain whereas those provided by $\mathcal{A}_i'$ are non-informative.

### 3.2. A Relevant Adaptation Set

It is critical that the sentences selected for a relevant adaptation set, $\mathcal{A}_r$, provide some new information about the test events that the training set, $\mathcal{S}_r$, does not have. We employed a simple but effective technique for this purpose. An initial training set is first constructed by randomly selecting 1 out-of-every 5 sentences in the original set, $\mathcal{O}$. The sentences in the other 80% of the set $\mathcal{O}$ are to be either selected as adaptation data sentences or added to the training set. In this work, those sentences that contain a test data event that the initial training data doesn’t have are selected for the adaptation set. The set of such sentences is further enriched by adding random sentences from $\mathcal{O}$ so that $\mathcal{A}_r$ has 10% of the training set.

### 3.3. Analysis of the Adaptation Sets

Consider Tables 1 and 2, where U, B and T denote the number of unigrams, bigrams and trigrams, respectively. Some crucial statistics are reported in Table 1 for $\mathcal{A}_i$, and in Table 2 for $\mathcal{A}_r$. A comparison of Tables 1 and 2 reveals several features of the two adaptation sets:

- The number of events in the data sets is shown in the first column. As expected, the number of events increases with the size of the set. A comparison of the first columns of Tables 1 and 2 reveals that the total number of events in same-size data sets of the two setups are very similar.

- The number of events in $\mathcal{A}_i' + \mathcal{S}$ is shown in the second column. Again, the number of events increases with the size of the set. Upon comparing the second columns of Tables 1 and 2, we see that having additional adaptation data, whether $\mathcal{A}_i$ or $\mathcal{A}_r$, results in data sets $\mathcal{S}_i + \mathcal{A}_i'$ or $\mathcal{S}_r + \mathcal{A}_r'$, with similar number of events. This suggests that the two sets provide similar number of n-gram events. The distinction is just that the events provided by $\mathcal{A}_r$ are informative about the test domain whereas those provided by $\mathcal{A}_i$ are non-informative.

- The number of test events that remains unseen in $\mathcal{S} + \mathcal{A}_r'$ is shown in the third column. As explained earlier, $\mathcal{A}_r$ is constructed such that it contains many test data events that $\mathcal{S}_r$ does not have. Hence, larger subsets of $\mathcal{A}_r$ leaves less test events unseen. As listed in the last row of the third column, there are 137 test bigrams and 854 test trigrams that were not seen in $\mathcal{O}$. By looking at the first row of third column in Table 1, we see that there are 144 test bigrams and 886 test trigrams that were not seen in $\mathcal{S}_i$. Hence, $\mathcal{A}_r$ provides additional 7 test bigrams and 32 test trigrams that were not seen in $\mathcal{S}_r$. Upon a similar comparison, we conclude that $\mathcal{A}_r$ provides additional 166 test bigrams and 510 test trigrams that were not seen in $\mathcal{S}_r$. Consider using only two-fifth of the available adaptation data instead of using no adaptation data, i.e. $\mathcal{A}_r' = \frac{2}{5} \mathcal{A}_r$. As shown in the second row of the third column in Table 1, when $\frac{2}{5} \mathcal{A}_r$ is used, only 2 test bigrams and 13 test trigrams that were not seen in $\mathcal{S}_r$ are added. On the other hand, when $\frac{2}{5} \mathcal{A}_r$ is used, 119 test bigrams and 310 test trigrams that were not seen in $\mathcal{S}_r$ are added.

<table>
<thead>
<tr>
<th>Table 1: Data statistics for the irrelevant adaptation set, $\mathcal{A}_i$</th>
<th>Table 2: Data statistics for the relevant adaptation set, $\mathcal{A}_r$</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Set</strong></td>
<td><strong>Total Events</strong></td>
</tr>
<tr>
<td><strong>in the Set</strong></td>
<td><strong>in $\mathcal{A}_i' + \mathcal{S}_i$</strong></td>
</tr>
<tr>
<td><strong>S_i</strong></td>
<td>4,985</td>
</tr>
<tr>
<td><strong>B</strong></td>
<td>1,556,766</td>
</tr>
<tr>
<td><strong>T</strong></td>
<td>8,519,647</td>
</tr>
<tr>
<td><strong>A_i' = \frac{2}{5} \mathcal{A}_i</strong></td>
<td>4,985</td>
</tr>
<tr>
<td><strong>B</strong></td>
<td>1,556,766</td>
</tr>
<tr>
<td><strong>T</strong></td>
<td>8,519,647</td>
</tr>
<tr>
<td><strong>A_r' = \frac{2}{5} \mathcal{A}_r</strong></td>
<td>4,985</td>
</tr>
</tbody>
</table>

\[ \mathcal{S}_i \subset \mathcal{S}_i' \subset \mathcal{S}_i'' \subset \cdots \subset \mathcal{A}_i \]\n
\[ \mathcal{S}_r \subset \mathcal{S}_r' \subset \mathcal{S}_r'' \subset \cdots \subset \mathcal{A}_r \]
4. Experimental Results

We performed experiments to illustrate the advantage of the proposed LM adaptation framework over ML-based language modeling. In our experiments, we set $\epsilon = 1$, $\gamma = 0$, and $\rho = 1$, and assume the same acoustic model set is used. We employed Good-Turing smoothing, and Laplace smoothing when a node has only one child node.

4.1. Performance of ML Models

The first set of experiment results in the upper part of Table 3 are obtained when maximum likelihood (ML) models were trained on the training data or on a subset of the adaptation data. The WERs were 5.04% and 6.63% when $S_i$ and $S_r$ training data sets were used, respectively. The WERs were 9.12% and 7.86% when one fifth of $A_i$ and $A_r$ were used, respectively. One-fifth of $A_r$ gave better results since it is richer in terms of test data events. In both cases, the WER reduced with the size of the adaptation set. When the entire $A_i$ or $A_r$ were used, the WER reduced to 7.23% and 5.17%, respectively.

The second set of experimental results in the bottom part of Table 2 were obtained when ML models were trained on training data plus subsets of the adaptation data. Since every subset of $A_r$ is very informative about the test domain, we observed a significant improvement in WER when larger subsets of $A_r$ were used. However, no significant change was observed in the ASR performance when $A_i$ was used.

4.2. Performance of Structural Bayesian LM Adaptation

The performance of the proposed LM adaptation framework is compared to that of the ML modeling in Figure 2. The curves marked with “□” and “□□” are obtained when $A_i$ and $S_r$ are used to find the ML estimates and to perform LM adaptation, respectively. The curves marked with “×” and “××” are obtained when $A_i$ and $S_i$ are used to find the ML estimates and to perform LM adaptation, respectively. The starting points for the LM adaptation experiments were obtained using the prior distribution probabilities only, i.e. with no adaptation.

Several observations follow from Figure 2. First of all, using $A_r$, for SMAP LM adaptation purpose instead of additional data for ML modeling uniformly reduces WER. It is remarkable that LM adaptation using the entire $A_r$ reduces WER to 4.73%, which is relatively 5.6% lower than what is obtained in all other three cases. Furthermore, a well-trained structured LM as prior information and re-estimating the parameters using an $A_r$ produced better results than training a model from scratch using $A_r$. On the other hand, there is no difference be-

<table>
<thead>
<tr>
<th>Data set</th>
<th>Coverage</th>
<th>$\hat{\alpha}_i$ and $S_i$</th>
<th>$\hat{\alpha}_r$ and $\omega_r$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$S_i$</td>
<td>90%</td>
<td>5.04</td>
<td>6.63</td>
</tr>
<tr>
<td>$S_i+\frac{1}{5}A_i$</td>
<td>2%</td>
<td>9.12</td>
<td>7.86</td>
</tr>
<tr>
<td>$S_i+\frac{4}{5}A_i$</td>
<td>4%</td>
<td>8.41</td>
<td>6.84</td>
</tr>
<tr>
<td>$S_i+\frac{6}{5}A_i$</td>
<td>6%</td>
<td>7.55</td>
<td>5.90</td>
</tr>
<tr>
<td>$S_i+\frac{8}{5}A_i$</td>
<td>8%</td>
<td>7.27</td>
<td>5.42</td>
</tr>
<tr>
<td>$S_i+A_i$</td>
<td>10%</td>
<td>7.23</td>
<td>5.17</td>
</tr>
<tr>
<td>$S+S_i+\frac{1}{5}A_i$</td>
<td>92%</td>
<td>5.01</td>
<td>6.35</td>
</tr>
<tr>
<td>$S+S_i+\frac{4}{5}A_i$</td>
<td>94%</td>
<td>5.03</td>
<td>5.98</td>
</tr>
<tr>
<td>$S+S_i+\frac{6}{5}A_i$</td>
<td>96%</td>
<td>5.04</td>
<td>5.79</td>
</tr>
<tr>
<td>$S+S_i+\frac{8}{5}A_i$</td>
<td>98%</td>
<td>5.06</td>
<td>5.40</td>
</tr>
<tr>
<td>$S+S_i+A_i$</td>
<td>100%</td>
<td>5.01</td>
<td>5.01</td>
</tr>
</tbody>
</table>

Figure 2: Using $A_r$ for SMAP LM adaptation purpose instead of additional data for ML modeling uniformly reduces WER.

5. Conclusions and Future Work

In this paper, we described a language modeling and adaptation framework based on structural Bayesian estimation. In the proposed scheme, each $n$-gram event, $\omega_1, ..., \omega_n$, is embedded in a tree structure. Within this structural framework, each node has an associated hyper-parameter representing the prior information, and a count representing the number of observation in the domain-specific data. Our experimental results verified that the proposed LM adaptation framework results in 5.6% relative improvement over the ML model.

This line of research is being furthered by incorporating the forgetting factor into the hyper-parameter estimation process. The forgetting factor enables the children nodes to inherit information from their parent nodes. This eventually provides tools to remedy the data-sparse-ness problem especially when high order $n$-grams are used. The proposed SMAP LM adaptation is also easily extendible to the class-based SMAP adaptation.

6. Acknowledgment

The second author thanks National Science Council, Taiwan for supporting his visiting research at Georgia Institute of Technology. The authors also thank Dr. Marco Siniscalchi for sharing his ASR engine and experience on WSJ corpus.

7. References