A Hierarchical Bayesian Approach for Semi-supervised Discriminative Language Modeling

Yik-Cheung Tam\textsuperscript{1}, Paul Vozila\textsuperscript{2}

Dragon Research, Nuance Communications Inc.
\textsuperscript{1}505 First Avenue South, Suite 700, Seattle, WA 98104, USA
\textsuperscript{2}One Wayside Road, Burlington, MA 01803, USA
\{wilson.tam,paul.vozila\}@nuance.com

Abstract
Discriminative language modeling provides a mechanism for differentiating between competing word hypotheses, which are usually ignored in traditional maximum likelihood estimation of N-gram language models. Discriminative language modeling usually requires manual transcription which can be costly and slow to obtain. On the other hand, there are vast amount of untranscribed speech data on which offline adaptation technique can be applied to generate pseudo-truth transcription as an approximation to manual transcription. Viewing manual and pseudo-truth transcriptions as two domains, we perform hierarchical Bayesian domain adaptation on discriminative language models sharing a common prior model. Domain-specific and prior models are estimated jointly using training data. In the N-best list rescoring experiment, hierarchical Bayesian domain adaptation has yielded better recognition performance than the model trained only on manual transcription, and seems robust against inferior prior.

Index Terms: Hierarchical Bayesian domain adaptation, Discriminative language modeling, semi-supervised learning

1. Introduction
Discriminative N-gram language modeling (LM) \cite{1,2,3} provides a mechanism to correct recognition errors made by a speech recognizer, which is not modeled by traditional maximum likelihood estimation of N-gram language models. In general, discriminative N-gram language modeling requires manual transcription to differentiate among word hypotheses in an N-best list for effective learning. Usually, obtaining manual transcription is costly and slow, creating a hurdle for the development of discriminative language modeling. On the other hand, it is significantly easier to obtain a vast amount of untranscribed speech data from a deployed speech recognition system covering a variety of speech recognition errors from the field. With the untranscribed speech data, we could obtain pseudo-truth transcription as a proxy for manual transcription by leveraging decoding and adaptation techniques on parameters that are not practical for real-time automatic speech recognition. Although pseudo-truth transcription is inferior to manual transcription, it will generally be better than the field recognition in terms of transcription accuracy. In this paper, we investigate leveraging the abundant pseudo-truth transcription with a small amount of manual transcription for improved discriminative language modeling.

Viewing manual and pseudo-truth transcription as two “domains”, we cast discriminative language modeling as domain adaptation. Since the N-gram features in manual and pseudo-truth transcription are correlated and highly overlapping, domain-specific feature parameters share a common prior distribution leading to Hierarchical Bayesian formulation \cite{4} or EasyAdapt \cite{5}. These approaches estimate domain-specific and prior models jointly in contrast to fixing the prior model and then estimating a target model as two steps \cite{6}. We show that by setting the hyper-parameters for regularization differently, Hierarchical Bayesian domain adaptation leads to different training scenarios such as independent model training per domain, training on target domain with a fixed out-of-domain prior, and the joint training scheme which gives the best performance in our evaluation.

Related work includes leveraging unsupervised speech data for conditional entropy regularization of maximum entropy language models \cite{7} to minimize the uncertainty of word hypotheses in a word lattice generated from an untranscribed speech utterance. Recent advancement of discriminative language modeling includes using pseudo-negative hypotheses \cite{8}, cohort set \cite{9}, and simulated confusion sets in machine translation \cite{10}.

We organize the paper as follows: In Section 2, we review discriminative language modeling including feature selection and model training. In Section 3, we describe Hierarchical Bayesian domain adaptation for semi-supervised discriminative language modeling. In Section 4, we evaluate our approach on different training scenarios. In Section 5, we conclude our work.
2. Review of discriminative language modeling

Traditional language modeling focuses on \( P(W) \) where \( W \) denotes a word sequence, while discriminative language modeling emphasizes \( P(W|X) \) where \( X \) denotes an input speech utterance. In other words, discriminative language modeling takes into account feedback from a speech recognizer. The goal is to differentiate correct word hypotheses from incorrect ones generated from the speech recognizer. Discriminative language modeling employs a feature-based exponential model:

\[
P(W|X; \Lambda) \approx \frac{e^{\sum_j \lambda_j F_j(X,W)}}{\sum_{Y \in \text{Gen}(X)} e^{F_j(X,Y)}} \tag{1}
\]

where \( \text{Gen}(X) \) denotes a set of word hypotheses applied to \( X \). The feature functions \( \{ F_j(X,W) \} \) takes an acoustic input \( X \) and word sequence \( W \) and return a real value. Typically, feature functions like \( F_{\text{amscore}}(X,W) \), \( F_{\text{lmscore}}(\cdot,W) \), and \( F_{\text{wip}}(\cdot,W) \) corresponding to the total acoustic score, the total language model score and the word insertion penalty, respectively, are augmented with the word N-gram count features \( F_{<w_1,...,w_n>}(\cdot,W) \) derived from the N-best lists. Computing the normalization term in Eqn 1 is not required during testing as it has no effect on ranking word hypotheses.

2.1. N-gram feature selection

A naive approach for N-gram feature selection to take all N-grams occurred in training lattices or N-best lists. However, this is impractical as this will lead to millions of features. To reduce the number of features, we use the perceptron algorithm for feature selection [3]. The idea is that if an N-gram feature is never updated during perceptron training, this feature is considered not useful for discriminating among word hypotheses. The perceptron algorithm is outlined below:

- **Initialization**: \( \Lambda = \{ \lambda_j \} = 0 \)
- **Decode** \( X \) and generate N-best list \( \{ \text{Gen}(X) \} \)
- **Align** N-best hypotheses with the reference to obtain the number of recognition errors.
- **If** the first-best hypothesis makes more errors than the oracle hypothesis \( W^* \), perform the weight update:

\[
\lambda_j^{(\text{new})} = \lambda_j^{(\text{old})} + F_j(X,W^*) - F_j(X,W_{1\text{best}}) \quad \forall j \tag{2}
\]

2.2. Training algorithm

Although one could directly use the trained weights from the voted perceptron algorithm, this is suboptimal compared to maximizing the conditional log-likelihood of the training data \( T \). After feature selection, we employ the conditional random field (CRF) [11] with gradient-ascent style parameter updates. To avoid over-fitting, we use a zero-mean Gaussian prior to penalize large feature weights so that they are regularized towards the trivial zero solution. The regularized conditional log-likelihood for CRF training is:

\[
L(\{ X, W^* \}; \Lambda) = \log \left( \frac{\prod_{<X,W^*> \in T} P(W^*|X; \Lambda)}{\sum_{<X,W^*> \in T} P(W^*|X; \Lambda)} \right) - \frac{|\Lambda|^2}{2\sigma^2} + \text{constant}
\]

3. Hierarchical Bayesian domain adaptation

In the machine learning community, Hierarchical Bayesian domain adaptation, a generalized version of EasyAdapt, is shown to be effective for domain adaptation for classifiers [5], maximum entropy language modeling [12] etc. The success is due to the co-regularization of domain models sharing a common prior, which is also estimated from data. Therefore, the prior and the domain models have mutual influence on each other. Empirically, Hierarchical Bayesian domain adaptation is effective when the domains are similar. For semi-supervised discriminative training, we treat manual transcription and pseudo-truth transcription as two “domains” and perform domain adaptation. However, our goal is to create a better target model for manual transcription. After training, the pseudo-truth model and the prior model are discarded and only use the trained manual transcription domain model for N-best rescoring.

With \( K \) domains, Hierarchical Bayesian domain adaptation maximizes the conditional log-likelihood of training data from \( K \) domains:

\[
\sum_{k=1}^{K} L(\{ X_k, W^*_k \}; \Lambda_k) - \sum_{k=1}^{K} \frac{|\Lambda_k - \Lambda|}{\sigma^2_k} - \frac{|\Lambda|}{\sigma^2} \tag{3}
\]

where \( \sigma^2_k \) and \( \sigma^2 \) are the hyper-parameters for regularization. Domain-specific models are regularized towards the prior model, which is regularized in turn towards zero. Parameter estimation is performed using gradient-based optimization of this convex function.
3.1. Prior and hyper-parameters

To understand the property of the prior model, we differentiate Eqn 3 w.r.t $\Lambda_k$ and set the derivative to zero:

$$\sum_{k=1}^{K} \Lambda_k - \Lambda_* = \frac{\Lambda_* - \Lambda_*}{\sigma^2_k} = 0$$  \hspace{1cm} (4)

$$\implies \Lambda_* = \sum_{k=1}^{K} w_k \cdot \Lambda_k$$  \hspace{1cm} (5)

where $w_k = \frac{1}{\sigma^2_k + \sum_{k'=1}^{K} \frac{1}{\sigma^2_{k'}}}$  \hspace{1cm} (6)

The prior model in Eqn 5 is a linear combination of domain-specific models with weight $w_k$ inversely proportional to the domain-specific hyper-parameters $\sigma^2_k$. This relationship offers some insight into manipulating the prior to achieve the purpose of maximizing the performance on the target domain. To understand the effect of the hyper-parameters, we assume there is only $K=2$ domains with the first and second being the target and non-target domains respectively. Below are the different training scenarios:

1. When $\sigma_1$ and $\sigma_2$ tend to zero, domain-specific models fall back to the prior model.

2. When $\sigma_1$ and $\sigma_2$ tend to infinity, the prior model has no effect to influence the domain-specific models, resulting in independent training per domain.

3. When $\sigma_2$ is arbitrarily smaller than $\sigma_1$ and $\sigma_*$, the prior model is mainly composed of the non-target model implied from Eqn 6.

4. When $\sigma_1 = \sigma_2 = \sigma_*$, Hierarchical Bayesian domain adaptation is equivalent to EasyAdapt which assumes equal importance per domain [4].

Since we are only interested in the performance of the target domain, i.e. manual transcription, it may not be appropriate to assume all domains to be equally important. Assuming that the non-target domain is helpful to the target domain, one heuristic is to assign a higher weight for the non-target domain in forming the prior in Eqn 5. The argument is that if the prior model is very similar to the target model, such “prior” will be less informative. Therefore, we enforce that $w_2 > w_1$ implying $\sigma_2 < \sigma_1$. Since the zero prior is less informative than all domain-specific models, $w_*$ should be made the smallest. Finally, we arrive at the heuristic to construct a prior model: $w_2 > w_1 > w_* \implies \sigma_* > \sigma_1 > \sigma_2$. Since there is a direct mapping between interpolation weights and hyper-parameters, it is usually more intuitive to tune on the interpolation weights and convert them back to hyper-parameters using Eqn 6, assuming $\sigma_*$ is fixed to some appropriate value.

3.2. Joint discriminative N-gram LM training

Similar to the training procedure described in Section 2, we perform N-gram feature selection independently per domain. Then we use the merged N-gram features plus the acoustic and word penalty features for joint model training. We substitute Eqn 5 into Eqn 3 to ensure that the linear combination constraint always holds during training. The partial derivative of the regularized conditional log-likelihood w.r.t $\Lambda_j$ is:

$$\frac{\partial L_j(.)}{\partial \Lambda_j} - \sum_{k=1}^{K} \frac{(\Lambda_k - \Lambda_*) \cdot (\delta(k, j) - w_j)}{\sigma^2_k} - \frac{w_j \cdot \Lambda_*}{\sigma^2_*}$$  \hspace{1cm} (7)

We perform gradient-based optimization. After training, we only keep the target model for N-best rescoring.

4. Experimental setup

We evaluated semi-supervised discriminative language modeling on an in-house speech recognition task using a modern ASR system. The training corpora consisted of a random sample of 60-hour manual transcription and 3000-hour pseudo-truth transcription using wider decoding beams, and offline acoustic and language model adaptation. There were two training domains: (1) manual transcription; (2) pseudo-truth transcription. The heldout set consisted of 10-hour manual transcription randomly drawn from the same pool of manually transcribed data. The baseline was a 4-gram language model with Kneser-Ney smoothing. Since the baseline language model contained pseudo-truth data, we built a modified model excluding the sample for discriminative language modeling. We used the modified model to generate realistic lattices. We applied two-pass decoding on the manual transcription and pseudo-truth transcription to generate 1000-best lists per utterance. Due to data sparsity, only unigram and bigram features were extracted from the N-best lists. The number of features from the target domain and all domains were 40k and 637k respectively after feature selection.

We compared the baseline 4-gram language model performance with discriminative language models that were trained with different scenarios: (a) 60-hour manual truth; (b) 3000-hour pseudo-truth (PT); (c) Same as scenario (a) but use the PT model as a fixed prior; (d) Hierarchical Bayesian domain adaptation with various regularization settings. In all regularization cases, $\sigma_*$ was fixed according to prior experience.

4.1. Rescoring results

Table 1 shows the number of error reduction and relative word error rate reduction compared to the baseline language model. The small amount of manual transcription for discriminative training helped reduce the number of errors relatively by 1.13%. On the other hand,
the pseudo-truth transcription in scenario (b) helped reduce the number of errors with similar degree. Therefore, combining the manual and pseudo-truth transcription helped, yielding a relative word error rate reduction by 1.75%. Performance of EasyAdapt is better than training only on the manual transcription, but is slightly worse than simple combination in scenario (c). Assuming the equal importance of target and non-target domains may be too strong, i.e. $\sigma_3 : \sigma_1 : \sigma_2 = 1 : 1 : 1$. With the heuristic, we increased the contribution of the non-target model to the prior via decreasing $\sigma_2$ and holding other hyper-parameters fixed, i.e. $\sigma_3 : \sigma_1 : \sigma_2 = 1 : 1 : 0.01$. This yielded slightly better performance than EasyAdapt. In scenario (d3), we increased $\sigma_1$ so that the target model has more freedom to depart from the prior. As expected, the resulted model yielded almost identical performance as independent training using manual transcription in scenario (a), since the prior had weaker influence. Therefore, we decreased $\sigma_1$ while keeping others fixed. With $\sigma_3 : \sigma_1 : \sigma_2 = 1 : 0.1 : 0.01$, Hierarchical Bayesian domain adaptation gave the best performance with 2.12% relative reduction in word error rate compared to the baseline language model.

Table 2 shows the performance of discriminative language modeling on the 37.8-hour test set. Unlike the performance on the heldout set, pseudo-truth transcription gave slight degradation compared to the baseline language model. The degradation may be attributed to mismatch between the pseudo-truth and manual transcription in the test set. The result also implies that pseudo-truth transcription leads to a less useful prior. Therefore, combining the target and non-target models either using a fixed pseudo-truth prior or Hierarchical Bayesian domain adaptation did not offer further benefit. However, both approaches seem to be robust against the mismatch. Moreover, hierarchical Bayesian domain adaptation still offered slight performance gain.

5. Conclusions

We presented a semi-supervised approach for discriminative language modeling, leveraging a large amount of untranscribed speech data via hierarchical Bayesian domain adaptation. Experimental results showed that hierarchical Bayesian domain adaptation has given slightly better performance than using a fixed pseudo-truth prior. Setting the hyper-parameters properly for regularization is crucial since different hyper-parameters values could lead to different training scenarios. Since the prior model is a linear combination of domain-specific models in hierarchical Bayesian domain adaptation, having the prior model to bias towards the non-target domain has resulted in more effective learning than EasyAdapt where all domains are assumed equally important. Relevance of non-target domain towards the test condition is crucial to form a good prior model. Even if the prior model is inferior, hierarchical Bayesian domain adaptation seems robust against it. In the future, we will investigate the domain mismatch issue.

6. References


---

Table 1: Word error rate reduction (WERR) relative to the baseline language model on the heldout set.

<table>
<thead>
<tr>
<th>Scenario</th>
<th># reduced error</th>
<th>Rel. WERR (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>baseline 4-gram LM</td>
<td>0</td>
<td>-</td>
</tr>
<tr>
<td>(a) 60hr manual truth</td>
<td>59</td>
<td>1.13%</td>
</tr>
<tr>
<td>(b) 3khr pseudo-truth (PT)</td>
<td>56</td>
<td>1.07%</td>
</tr>
<tr>
<td>(c) 60hr w/ fixed PT-prior</td>
<td>59</td>
<td>1.13%</td>
</tr>
<tr>
<td>(d1) EasyAdapt (1:1:1)</td>
<td>79</td>
<td>1.51%</td>
</tr>
<tr>
<td>(d2) HierBayes (1:1:0.01)</td>
<td>89</td>
<td>1.70%</td>
</tr>
<tr>
<td>(d3) HierBayes (1:1:0.01)</td>
<td>58</td>
<td>1.11%</td>
</tr>
<tr>
<td>(d4) HierBayes (1:0:1:0.01)</td>
<td>111</td>
<td>2.12%</td>
</tr>
</tbody>
</table>

Table 2: Word error rate reduction (WERR) relative to the baseline language model on the test set.

<table>
<thead>
<tr>
<th>Scenario</th>
<th># reduced error</th>
<th>Rel. WERR (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>baseline 4-gram LM</td>
<td>0</td>
<td>-</td>
</tr>
<tr>
<td>(a) 60hr manual truth</td>
<td>253</td>
<td>0.76%</td>
</tr>
<tr>
<td>(b) 3khr pseudo-truth (PT)</td>
<td>-298</td>
<td>-0.8%</td>
</tr>
<tr>
<td>(c) 60hr w/ fixed PT-prior</td>
<td>284</td>
<td>0.76%</td>
</tr>
<tr>
<td>(d) HierBayes (1:0:1:0.01)</td>
<td>-292</td>
<td>-0.8%</td>
</tr>
</tbody>
</table>