Adapting n-gram Maximum Entropy Language Models with Conditional Entropy Regularization

Ariya Rastrow, Mark Dredze, Sanjeev Khudanpur

Human Language Technology Center of Excellence
Center for Language and Speech Processing, Johns Hopkins University
Baltimore, MD USA
{ariya,mdredze,khudanpur}@jhu.edu

***Abstract*—Accurate estimates of language model parameters are critical for building quality text generation systems, such as automatic speech recognition. However, text training data for a domain of interest is often unavailable. Instead, we use semi-supervised model adaptation; parameters are estimated using both unlabeled in-domain data (raw speech audio) and labeled out of domain data (text.) In this work, we present a new semi-supervised language model adaptation procedure for Maximum Entropy models with n-gram features. We augment the conventional maximum likelihood training criterion on out-of-domain text data with an additional term to minimize conditional entropy on in-domain audio. Additionally, we demonstrate how to compute conditional entropy efficiently on speech lattices using first- and second-order expectation semirings. We demonstrate improvements in terms of word error rate over other adaptation techniques when adapting a maximum entropy language model from broadcast news to MIT lectures.**

I. Introduction

In many text generation systems, such as Automatic Speech Recognition (ASR) and Machine Translation, a language model is crucial for selecting reasonable outputs. For example, in ASR systems, the goal is to find a word string W with maximum a posteriori probability given the acoustics X:

$$\widehat{W} = \arg\max_W P(W|A) = \arg\max_W P(A|W)P(W)$$

The language model estimates an *a priori* probability $P(W)$ that the speaker will utter W. Accurately modeling these probabilities can have a significant impact on task performance.

One approach to estimating the LM's parameters relies on Maximum Entropy (MaxEnt) learning. In recent years, MaxEnt learning has become a popular approach for estimating language model parameters from a large corpus of text [1], [2]. MaxEnt models allow for the inclusion of arbitrary features, providing a unified framework for combining a range of different dependencies [3]. Additionally, MaxEnt learning provides a well formed model to which new machine learning techniques can be naturally applied. In contrast, n-gram language models must satisfy back-off constraints and ensure that the model parameters (conditional probabilities) sum to one. MaxEnt models often yield better performance compared to standard n-gram models [3], [2].

No matter the training algorithm, language models must accurately model the language for the application domain of interest. Like other statistical models, the modeling assumption is that the training data (text) and the test data (speech audio) come from the same distribution. Presumably, the training data is an accurate representation of the test data. However, often times this is not the case. For many speech domains, we have plentiful training text from another domain (labeled data) but little or no in-domain language model training text. Instead, we have raw speech audio (unlabeled data) from the domain of interest. In this case, we seek to adapt a language model trained on the out-of-domain data to our target domain using what little in-domain resources are available. If we have a small amount of in-domain training text, we term this setting *supervised* adaptation. If there is only in-domain speech audio and no text, we require *semi-supervised* adaptation.

In this work, we focus on language models with n-gram features trained using MaxEnt and consider the more challenging and prevalent case of *semi-supervised* adaptation. We augment the standard training objective to include conditional entropy regularization on the in-domain speech audio. By minimizing the conditional entropy on unlabeled in-domain data, we encourage the model to prefer parameters that minimize class overlap in the target domain [4]. While conditional entropy regularization has been previously applied to different classification tasks [4], [5], [6], [7], we demonstrate how to compute this regularization efficiently on speech lattices using first- and second-order expectation semirings [8]. We compare our new training objective with a fully supervised adaptation method [9] as well as a semi-supervised method based on self-training [10]. We demonstrate improvements in the semi-supervised case and show that we are approaching the supervised setting.

We proceed as follows. In Section II we provide an overview of n-gram MaxEnt language models and their training procedures. Section III introduces our new objective function based on conditional entropy regularization for language model adaptation followed by a semiring algorithm for calculating the conditional entropy on unlabeled speech lattices. After an overview of related adaptation techniques, we present results on adapting from broadcast news to MIT lectures.

II. n-gram MaxEnt Language Modeling

Under the MaxEnt framework, the probability of word w given the history h has a log-linear form,

$$P_\Lambda(w|h) = \frac{1}{Z_\Lambda(h)} \exp\left(\sum_i \lambda_i \cdot f_i(w,h)\right). \quad (1)$$

$Z_\Lambda(h)$ is the normalization factor for the given history h,

$$Z_\Lambda(h) = \sum_{w' \in V} \exp\left(\sum_i \lambda_i \cdot f_i(w',h)\right). \quad (2)$$

$f_i(w,h)$ is the i-th feature function based on word w and history h. Each λ_i represents the weight of the corresponding feature and the set of feature weights Λ forms the parameters of the model, estimated during LM training. In the case of n-gram MaxEnt models, each n-gram corresponds to a single features. For instance, a bigram feature would take the form:

$$f_i(w,h) = \begin{cases} 1 & \text{if } w = a \text{ and } h \text{ ends in } b \\ 0 & \text{otherwise} \end{cases}$$

for some a and b. Typically, for an n-gram MaxEnt model the feature set and parameters are defined to include all the n-grams ($n = 1, 2, \cdots N$) seen in a training data.

A. Training Procedure

Supervised parameter estimation algorithms for n-gram MaxEnt LMs fit the training sentences using a Maximum Likelihood (ML) criterion. Let the training data $\mathbf{W} = \{\mathbf{w}_1, \mathbf{w}_2, \cdots \mathbf{w}_l\}$ be comprised of l sentences, and let n_j denote the number of words in sentence $\mathbf{w}_j$. The log-likelihood of the training corpus $\mathbf{W}$ using the MaxEnt language model parameters Λ can be written as,

$$\mathcal{L}(\mathbf{W};\Lambda) = \log P(\mathbf{W}) = \sum_{j=1}^{l} \sum_{i=1}^{n_j} \log P_\Lambda(w_i|h_i). \quad (3)$$

Maximizing $\mathcal{L}(\mathbf{W};\Lambda)$ yields trained model parameters

$$\hat{\Lambda} = \arg\max_\Lambda \mathcal{L}(\mathbf{W};\Lambda) = \arg\max_\Lambda \sum_{j=1}^{l} \sum_{i=1}^{n_j} \log P_\Lambda(w_i|h_i). \quad (4)$$

There are several approaches to maximizing this objective, such as Generalized Iterative Scaling (GIS) [11], or gradient based methods, such as L-BFGS [12]. In this work, we use gradient based optimization, specifically, Quasi-Newton L-BFGS. It is easy to show that the gradient of the log-likelihood with respect to the model parameters Λ can be calculated as,

$$\nabla_\Lambda \log P(\mathbf{W}) = \sum_{w,h} c(w,h) \cdot \Delta_\Lambda(w,h), \quad (5)$$

$$\text{where } \Delta_\Lambda(w,h) = \mathbf{f}(w,h) - \sum_{w'} P_\Lambda(w'|h)\mathbf{f}(w',h) \quad (6)$$

and $\mathbf{f}(w,h)$ is a vector which has value one for the activated n-gram features corresponding to (w,h) and zero elsewhere. Hence, the gradient is the difference Δ between the observed and expected n-gram features (over the MaxEnt distribution) summed over every event (w,h) weighted by its observed count $c(w,h)$ in the training data. In addition, an L_2 regularization term $||\Lambda||_2^2$ with weight γ is added to the objective function (Eq. 4) to prevent overfitting and provide smoothing [13]. γ is usually chosen empirically using a development set.

B. Hierarchical Training Technique

A significant disadvantage of MaxEnt LM training is the need to compute the normalizer (partition function) $Z_\Lambda(h)$ of (2), which must sum over *all possible* words w to achieve a valid probability. In the naive implementation of a MaxEnt LM, the complexity for computing normalization factors (and feature expectations) for a single iteration is $O(|H| \times |V|)$, where $|H|$ is the number of history tokens seen in $\mathbf{W}$ and $|V|$ is the vocabulary size, typically on the order of tens of thousands. We instead use the hierarchical training procedure introduced in [3] for nested and non overlapping features, e.g., n-gram features. The hierarchical training procedure reduces the complexity for calculating normalization factors and n-gram feature expectations to $O(\sum_{n=1}^{N} \#n\text{-grams})$, the same complexity as training the corresponding back-off n-gram LM (where we must compute the frequency for all seen n-grams.)

III. Language Model Adaptation

While n-gram language models are effective at learning a probability distribution that can explain a given corpus, they fail to assign realistic probabilities to new data that differ from training examples. In this case, new grammatical structures and previously unseen n-grams are estimated to be very unlikely, which can degrade system performance in new domains. For new domains without text training data, we seek to train on available out-of-domain text data and in-domain audio. We can draw on techniques from work in semi-supervised learning to facilitate *semi-supervised* adaptation. One such approach consistent with MaxEnt trained language models is conditional entropy regularization. We review this method and define a new language model adaptation objective.

A. Conditional Entropy Regularization

Consider a classification problem where $\mathbf{X}$ are the inputs (observations) and $\mathbf{Y}$ are the corresponding output (class) labels. The conditional entropy $H(\mathbf{Y}|\mathbf{X})$ is a measure of the average (expected) randomness in the probability distribution of class labels $\mathbf{Y}$ after observing the input $\mathbf{X}$.

$$H(\mathbf{Y}|\mathbf{X}) = E_\mathbf{X}[H(\mathbf{Y}|\mathbf{X} = x)] = -\int p(x) \left(\sum_y p(y|x) \log p(y|x)\right) dx \quad (7)$$

Conditional entropy measures the amount of the class overlap and can be related to classification performance through the well known *Fano's Inequality* [7], [14]. This inequality proves that $\mathbf{Y}$ can be estimated with low probability of error only if the conditional entropy $H(\mathbf{Y}|\mathbf{X})$ is small. Intuitively, class overlap indicates uncertainty about the example and by minimizing the entropy, we encourage the model to prefer parameters that minimize class overlap, thereby minimizing uncertainty. The trained low conditional entropy model will have a decision boundary that passes through low-density regions of the input distribution $p(\mathbf{X})$. For problems with

well separated classes, we can take advantage of unlabeled examples to find low-density regions [4].

Because conditional entropy minimizes label uncertainty, it has been used as a regularization term for unlabeled data in semi-supervised learning, e.g. it is used alongside the maximum likelihood criterion for semi-supervised training in classification [4]. Similar objectives have been used in [5] for training a Conditional Random Field (CRF) for identifying genes and proteins. Entropy minimization is used in [6] as an unsupervised non-parametric clustering method and is shown to result in a significant improvement over k-means, hierarchical clustering, and other clustering algorithms. Conditional entropy minimization is also used in [7] to estimate parameters for interpolating n-gram LMs in an unsupervised fashion.

B. Adaptation Objective Function

We introduce a new semi-supervised objective based on conditional entropy regularization for the adaptation of n-gram MaxEnt language models. We have $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, \cdots \mathbf{x}_{n_u}\}$, a set of n_u audio utterances from in-domain speech, for which we have no reference transcripts. Additionally, we have $\mathbf{T} = \{\mathbf{w}_1, \mathbf{w}_2, \cdots, \mathbf{w}_{n_l}\}$, a set of n_l sentences from out-of-domain data. Our goal is to use the out-of-domain text $\mathbf{T}$ (labeled) and the in-domain audio speech data $\mathbf{X}$ (unlabeled) to train a language model for recognizing in-domain speech.

As usual, we will maximize the log-likelihood $\mathcal{L}(\mathbf{T}; \Lambda)$ (Eq. 3) of the MaxEnt language model on the training text $\mathbf{T}$. Additionally, we will minimize the empirical conditional entropy $H_{\text{emp}}(\mathbf{W}|\mathbf{X}; \Lambda, \Gamma)$ of the posterior probability over word sequences $\mathbf{W}$ given the in-domain audio $\mathbf{X}$ imposed by MaxEnt language model parameters Λ and acoustic model parameters Γ.

By minimizing this term, we estimate parameters that result in unambiguous recognizer output by guiding the decision boundaries away from dense regions of the in-domain input distribution, resulting in *peaked/confident* recognizer output. Specifically the semi-supervised adaptation objective is

$$\max_{\Lambda} \left[\frac{1}{n_l} \mathcal{L}(T_{n_l}; \Lambda) - \gamma ||\Lambda||_2^2 - \sigma H_{\text{emp}}(\mathbf{W}|\mathbf{X}; \Lambda, \Gamma) \right]. \quad (8)$$

We include the standard L_2 regularizer $||\Lambda||_2^2$ with scale parameter γ, and the effect of conditional entropy is controlled by the scale parameter σ, which we tune on a very small amount of *transcribed* in-domain speech to minimize word error rate (WER).

The first part of this objective, the log-likelihood term, is unaffected by the unlabeled in-domain data and is computed as usual. The conditional entropy term depends only on the posterior probabilities of the in-domain unlabeled data,

$$H_{\text{emp}}(\mathbf{W}|\mathbf{X}; \Lambda, \Gamma) = -\sum_{i=1}^{n_u} \sum_{W} \frac{P_{\Lambda,\Gamma}(W|\mathbf{x}_i) \log P_{\Lambda,\Gamma}(W|\mathbf{x}_i)}{n_u}$$

where

$$P_{\Lambda,\Gamma}(W|\mathbf{x}_i) = \frac{P_\Gamma(\mathbf{x}_i|W)^\alpha \cdot P_\Lambda(W)}{\sum_{W'} P_\Gamma(\mathbf{x}_i|W')^\alpha \cdot P_\Lambda(W')} \quad (9)$$

is the posterior probability of the word sequence W given the audio according to the recognizer, α is the standard acoustic weight and the sum $\sum_{W'}$ runs over all possible word sequences given the audio $\mathbf{x}_i$.

Since almost any word sequence can have a non-zero probability given the speech audio, it is intractable to enumerate over all possible word sequences. Instead, we compute an approximation by restricting the word sequences to those present in the corresponding *lattice* $\mathbf{L}$:

$$\begin{aligned} H_{\text{emp}}(\mathbf{W}|\mathbf{X}; \Lambda, \Gamma) &\approx H_{\text{emp}}(\mathbf{W}|\mathbf{L}; \Lambda, \Gamma) \\ &= -\frac{1}{n_u} \sum_{i=1}^{n_u} \sum_{W \in \mathbf{L}_i} P_{\Lambda,\Gamma}(W|\mathbf{L}_i) \log P_{\Lambda,\Gamma}(W|\mathbf{L}_i). \quad (10) \end{aligned}$$

Including $H_{\text{emp}}(\mathbf{W}|\mathbf{L}; \Lambda, \Gamma)$ in the objective function requires computing the conditional entropy over the lattice, and its gradient w.r.t. the model parameters Λ. We provide an algorithm for doing so efficiently in Section IV.

As usual, the log-likelihood term and the L_2 regularizer are concave functions of Λ. However, the conditional entropy regularizer is non-convex, resulting in a non-convex objective, subject to local maxima. Therefore, initialization plays an important role in finding good solutions. In this work we first learn Λ using the fully supervised objective (omit conditional entropy), yielding a reasonable solution given the out-of-domain data. We then use these values to initialize the parameters of the full semi-supervised model, which modifies the parameters based on the in-domain data. Since we expect the optimal parameters to be similar for both domains, using the supervised objective yields parameters close to the best parameters on the in-domain data. We found this a more effective strategy than using the full model initially, since forcing minimum conditional entropy on in-domain data can yield degenerate solutions.

For clarity, we emphasize that there is no contradiction in minimizing the conditional entropy of a maximum entropy model. The choice of Λ maximizes the difference between (i) the *entropy of the language model* constrained to obeying n-grams frequencies extracted reliably from the out-of-domain text and (ii) the *conditional entropy of the recognizer output* given the in-domain acoustics. The former term prefers a maximally smooth model that is consistent with the training text, while the latter a maximally unambiguous transcript of the unlabeled speech.

IV. Conditional Entropy on Speech Lattices

In the previous section, we introduced a new conditional entropy regularizer $H_{\text{emp}}(\mathbf{W}|\mathbf{X}; \Lambda, \Gamma)$ for semi-supervised language model training (Eq. 8.) The regularizer required a summation over all possible word sequences, which we approximated by restricting this summation to word sequences that appear in the lattice $H_{\text{emp}}(\mathbf{W}|\mathbf{L}; \Lambda, \Gamma)$. The lattice is the direct output of the ASR system, a directed acyclic graph (DAG) with unique start and end nodes, nodes time-stamped with respect to the speech signal $\mathbf{x}$, and edges labeled with words w. Each path in the DAG from start to end corresponds

to a candidate time-aligned transcripts $W = w_1 w_2 ... w_n$ of $\mathbf{x}$. The lattice can be represented as a finite-state automata (FSA), a compact representation of the hypothesis space for a given speech signal $\mathbf{x}$. To optimize the objective (8), we must calculate the lattice conditional entropy (defined in Eq. 10) and its gradient with respect to the MaxEnt language model parameters Λ. We note, as an aside, that the lattices are generated using a regular (unadapted) n-gram LM.

Since the number of paths (hypotheses) in the lattice is very large, it would be computationally infeasible to compute the conditional entropy by enumerating all possible lattice paths and calculating their corresponding posterior probabilities. Instead we use the first- and second-order *semirings* [8] on the FSA-representation of the lattice along with the forward algorithm to calculate the conditional entropy and its gradient. We define the probability of a path d as $p(d) = P_{\Lambda,\Gamma}(W(d)|\mathbf{L})$ where d is a path in $\mathbf{L}$ and $W(d)$ is the word sequence corresponding to the path. Therefore, the conditional entropy $H(p)$ of all the paths in a lattice can be written as

$$\begin{aligned} H(p) &= -\sum_{d\in\mathbf{L}} p(d)\log p(d) = -\sum_{d\in\mathbf{L}} \frac{p'(d)}{Z(\mathbf{L})}\log\frac{p'(d)}{Z(\mathbf{L})} \\ &= \log Z(\mathbf{L}) - \frac{1}{Z(\mathbf{L})}\sum_{d\in\mathbf{L}} p'(d)\log p'(d) \\ &= \log Z(\mathbf{L}) - \frac{\bar{r}}{Z(\mathbf{L})} \end{aligned} \tag{11}$$

where $p'(d) = P_\Gamma(\mathbf{x}|W(d))^\alpha \cdot P_\Lambda(W(d))$ is the unnormalized score for the path d, and $Z(\mathbf{L}) = \sum_{d'\in\mathbf{L}} p(d')$ is the marginal probability of the lattice $\mathbf{L}$. Using (11) it is easy to show that,

$$\begin{aligned} \nabla_\Lambda H_{\text{emp}}(\mathbf{W}|\mathbf{L};\Lambda,\Gamma) &= \nabla_\Lambda H(p) \\ &= \frac{\nabla_\Lambda Z(\mathbf{L})}{Z(\mathbf{L})} - \frac{Z(\mathbf{L})\nabla_\Lambda\bar{r} - \bar{r}\nabla_\Lambda Z(\mathbf{L})}{Z(\mathbf{L})^2} \end{aligned} \tag{12}$$

Hence it suffices to calculate $\mathbf{S} = \langle Z(\mathbf{L}), \bar{r}, \nabla_\Lambda Z(\mathbf{L}), \nabla_\Lambda \bar{r}\rangle$ on the lattices to get the conditional entropy and its gradient.

It is shown in [8] that if we define a second-order semiring $\mathbf{K} = \langle k_e, \otimes, \oplus, \mathbf{0}, \mathbf{1}\rangle$ with a 4-tuple weight k_e equal to $\langle p_e, \log p_e, \nabla_\Lambda p_e, (1+\log p_e)\nabla_\Lambda p_e\rangle$, additive operation $\oplus$ and multiplicative operation $\otimes$ as shown in Table I, and the score p_e on each arc of the lattice defined as

$$p_e = (p_\Gamma)^\alpha \cdot p_\Lambda$$

where p_Γ and p_Λ are the acoustic and language model scores respectively, then running the forward algorithm on the lattice we will yield $\mathbf{S}$ as the forward weight of the last node.

For an arc with word-history pair (w,h), p_Λ using the n-gram MaxEnt formulation is defined in (1). Therefore, $\nabla_\Lambda p_e$ is calculated as,

$$\begin{aligned} &\nabla_\Lambda p_e = p_e \cdot (\nabla_\Lambda \log p_e) = p_e \cdot \nabla_\Lambda \log p_\Lambda \\ &= p_e \cdot \nabla_\Lambda \log\left[\frac{1}{Z_\Lambda(h)}\exp\left(\sum_i \lambda_i \cdot \mathbf{f}_i(w,h)\right)\right] \\ &= p_e \cdot [\mathbf{f}(w,h) - \nabla_\Lambda \log Z_\Lambda(h)] = p_e \cdot \Delta_\Lambda(w,h) \end{aligned} \tag{13}$$

where the first equality is due to the fact that $\nabla_\Lambda \log p_e = \frac{\nabla_\Lambda p_e}{p_e}$, and $\Delta_\Lambda(w,h)$ is from (5). Using the simplifications above, one may re-write the semiring weight as

$$k_e = \langle p_e,\ \log p_e,\ p_e\Delta_\Lambda(w,h),\ p_e(1+\log p_e)\Delta_\Lambda(w,h)\rangle$$

where again (w,h) is the corresponding word and history of the arc on which k_e is defined. After we compute k_e on each arc of the lattice, we run the forward algorithm using the operations shown in Table I to compute $\mathbf{S}$ and therefore, the conditional entropy and its gradient (Eqs. 11 and 12).

A. Efficient Implementation

The semiring weight k_e requires vectors in the third and fourth positions of size equal to the vocabulary ($\Delta_\Lambda(w,h)$ is a vector) and it would be computationally expensive, both in terms of time and memory, to calculate and save these vectors for all the arcs of the lattice. We discuss improvements that both speed up the calculations and also reduce memory usage.

We use the forward-backward speedup for second-order expectation semirings as explained in Section 4.4 of [8] [1] (in [8] the technique is referred to as inside-outside speedup). The key idea is to run the forward-backward algorithm (instead of only the forward algorithm) on first order semiring weights $k'_e = \langle p_e, \log p_e\rangle$ (instead of second order semiring weights k_e) and then iterate through the arcs of the lattice to update one global second order semiring weight. After visiting all the arcs the updated global second order semiring is shown to be equal to $\mathbf{S}$. In this approach, only 2 vocabulary size vectors are saved at any given time (for the arc that we visit), a dramatic reduction when compared to requiring $2N_\mathbf{L}$ such vectors, where $N_\mathbf{L}$ is the number of arcs in $\mathbf{L}$.

While this reduces the memory usage, we still need to compute $Z_\Lambda(h)$ and $\nabla_\Lambda \log Z_\Lambda(h)$ (the feature expectation part of $\Delta_\Lambda(w,h)$) for each arc in the lattice, making the algorithm's complexity on each lattice $O(N_\mathbf{L} \times |V|)$. We note that although each arc of the lattice corresponds to a (w,h) pair, $Z_\Lambda(h)$ and $\nabla_\Lambda \log Z_\Lambda(h)$ are only functions of the history h. Hence, we can reduce the complexity by caching the values of $Z_\Lambda(h)$ and $\nabla_\Lambda \log Z_\Lambda(h)$ for each history (instead of calculating them for each arc.) This will reduce the complexity to $O(|H| \times |V|)$ where $|H|$ is the number of unique histories seen in the lattice. It is worth mentioning that $|H|$ is usually 10 to 20 times smaller than $N_\mathbf{L}$ due to timing information encoded in the lattice nodes. The same history can happen in many different arcs with different timings. This caching results in saving $|H|$ vocabulary size vectors ($\nabla_\Lambda \log Z_\Lambda(h)$ is a vector). To summarize, the complexity of our algorithm is $O(|H| \times |V|)$ and we store $|H|$+2 vectors of size $|V|$.

Even after our improvements, the algorithm is still computationally expensive. However, this procedure is conducted only when we need to adapt a model for a new domain, especially for large vocabulary tasks. Once adapted, the trained n-gram LM can be used with complexity equal to standard MaxEnt LMs.

[1] We refer the reader to [8] for a detailed explanation of the algorithm.

Element	$\langle p, r, s, t \rangle$
$\langle p_1, r_1, s_1, t_1 \rangle \otimes \langle p_2, r_2, s_2, t_2 \rangle$	$\langle p_1 p_2, p_1 r_2 + p_2 r_1, p_1 s_2 + p_2 s_1, p_1 t_2 + p_2 t_1 + r_1 s_2 + r_2 s_1 \rangle$
$\langle p_1, r_1, s_1, t_1 \rangle \oplus \langle p_2, r_2, s_2, t_2 \rangle$	$\langle p_1 + p_2, r_1 + r_2, s_1 + s_2, t_1 + t_2 \rangle$
$\mathbf{0}$	$\langle 0, 0, 0, 0 \rangle$
$\mathbf{1}$	$\langle 1, 0, 0, 0 \rangle$

TABLE I
SECOND-ORDER SEMIRING: DEFINING *multiplicative* AND *additive* OPERATIONS FOR SECOND-ORDER SEMIRINGS.

V. RELATED WORK

There have been several approaches to language model adaptation. [9] proposed a simple and general MaxEnt (log-linear) adaptation algorithm for settings in which a limited amount of in-domain data is available for training, which has been successfully used for supervised LM adaptation [2]. The parameters of an out-of-domain model are used as priors, expressed as an L2-regularizer, for training of the in-domain model. For language model adaptation, the objective becomes,

$$\hat{\Lambda} = \arg\max_{\Lambda} \mathcal{L}(\mathbf{W}_T; \Lambda) - \gamma ||\Lambda - \Lambda_O||^2 \quad (14)$$

where $\mathcal{L}(\mathbf{W}_T; \Lambda)$ is the log-likelihood on the in-domain training text $\mathbf{W}_T$ and Λ_O are the MaxEnt parameters trained on the out-of-domain data. γ is tuned on in-domain development data. [2] showed that using the out-of-domain model as a prior for in-domain parameter estimation resulted in better performance than interpolating out-of-domain and in-domain models when some labeled in-domain data was available for training. We call this method `Prior` and use it as a supervised baseline for comparison against our semi-supervised technique.

The most relevant semi-supervised adaptation technique is *self-training*, where the 1-best output of the ASR system on in-domain speech data is used as the training text for re-estimating parameters of the MaxEnt model [10]. This setup can be augmented with the above approach as follows,

$$\Lambda_{\text{new}} = \arg\max_{\Lambda} \mathcal{L}(\mathbf{W}^*_D(\Lambda_{\text{old}}); \Lambda) - \gamma ||\Lambda - \Lambda_{\text{old}}||^2 \quad (15)$$

$W^*_D(\Lambda_{\text{old}})$ is the 1-best output of the ASR system for the in-domain speech data after rescoring with an n-gram MaxEnt model with parameters Λ_{old}. Therefore, we select new parameters to both maximize the probability of the 1-best ASR output and to minimize the change from the old parameters. We begin by setting $\Lambda_{\text{old}} = \Lambda_S$ and then iterate, producing 1-best output and training a new model, each time using the parameters from the previous iteration. The number of iterations is tuned on the development data. We call this method `Self-Train` and use it as a semi-supervised baseline.

While our method is more computationally intensive than self-training, it has the advantage of being *discriminative*. The conditional entropy regularizer is calculated on the lattice, which captures the acoustic confusions and the language model parameters are adjusted to reduce these confusions.

VI. EVALUATION

A. Experimental Setup

For our evaluation we consider the adaptation of a 3-gram MaxEnt language model from broadcast news to the MIT lectures domain. While the broadcast news domain is well structured with many rare proper nouns, MIT lectures are much more fluid with many technical terms. There are plentiful broadcast news text data but little text data for MIT.

We used the 2007 IBM speech transcription system for the GALE Distillation Go/No-go Evaluation [15]. The acoustic models are state of the art discriminatively trained models trained on Broadcast News (BN) Hub4 acoustic training data. We used the EARS BN03 closed captions corpus as the out-of-domain training text (BN training text). The ASR system's initial model is a pruned modified Kneser-Ney backoff 3-gram model, which we refer to as "Baseline 3-gram." The initial ASR lattices on the in-domain audio data are then rescored using a second language model. The second model is our 3-gram MaxEnt model, trained on the same EARS BN03 corpus. For this model there was no feature cut-off, resulting in about 17 million parameters ($|\Lambda_O| = 17M$). The L_2 regularizer's weight was $\gamma = 0.7$ based on tuning using the rt03 BN development set. We call refer to this model "MaxEnt 3-gram."

Our in-domain data comes from the MIT lectures corpus [16], which is split into an adaptation set – 5 hours or about 1800 utterance and 42K words – a development set – 0.5 hour – and an evaluation set – 2.5 hours or about 1K utterances and 20K words. For semi-supervised adaptation we use the 5 hours of speech audio from the adaptation set and for supervised adaptation we use the associated adaptation set transcripts.

To train our conditional entropy (CE) regularization model, we optimized the objective function (Eq. 8 and Eq. 10) by calculating the conditional entropy and its gradient on the MIT in-domain adaptation set speech lattices. The lattices were generated by the recognizer using the Baseline 3-gram language model. We initialize model parameters Λ to the parameters obtained by training MaxEnt 3-gram on the BN data (Λ_O). The regularization parameter was $\gamma = 0.7$ for both the MaxEnt 3-gram model and CE regularization. Optimizing on MIT development for WER yielded the CE regularization parameter $\sigma = 2.5$.

The `Self-Train` method was trained by starting from Λ_O and then iteratively optimizing Eq. 15. We stopped training after two iterations based on WER on the development set.

It is worth mentioning that we do not let the number of parameters increase in any of the adaptation methods.

B. Results

We evaluate our conditional entropy trained language model (CE Regularization) on the MIT corpus in terms of both perplexity and WER on the reference transcripts in the evaluation set. We compare our model to the two baseline out-of-domain

Adaptation Type	Language Model	Perplexity	WER
None	Baseline 3-gram	289	25.9
	MaxEnt 3-gram	278	25.7
Semi-Supervised	`Self-Train`	263	25.3
	CE Regularization	264	**24.9**
Supervised	`Prior`	221	**22.8**

TABLE II
PERPLEXITY AND WER RESULTS ON MIT EVALUATION DATA FOR DIFFERENT ADAPTATION METHODS. OUR SEMI-SUPERVISED APPROACH OF CONDITIONAL ENTROPY (CE) REGULARIZATION IMPROVES OVER THE OUT-OF-DOMAIN BASELINES AND PREVIOUS SEMI-SUPERVISED METHODS (`Self-Train`).

Adaptation Type	Language Model	Conditional Entropy
None	MaxEnt 3-gram	9.61
Semi-Supervised	`Self-Train`	7.34
	CE Regularization	6.48
Supervised	`Prior`	5.88

TABLE III
MEAN CONDITIONAL ENTROPY ON THE MIT ADAPTATION SET FOR EACH OF THE MAXENT LANGUAGE MODELS. OUR CE REGULARIZATION ACHIEVES THE LOWEST VALUES AMONG THE SEMI-SUPERVISED AND UNADAPTED MODELS. FURTHER REDUCTIONS BY THE SUPERVISED METHOD SUGGESTS THAT CONDITIONAL ENTROPY MINIMIZATION IS A SENSIBLE ADAPTATION OBJECTIVE.

language models ("Baseline 3-gram" and "MaxEnt 3-gram") as well as the semi-supervised `Self-Train` and supervised `Prior` methods. The `Prior` model's regularization parameter γ was tuned on the MIT development set, yielding $\gamma = 0.5$. Results for all five language models are reported in Table II.

We first observe that switching from standard n-gram language model training to MaxEnt training gives a 0.2 improvement in WER, which confirms previously published results that MaxEnt language models perform better [3], [2]. This can be attributed to either a better training procedure or because the Baseline 3-gram model is heavily pruned to facilitate lattice generation; there was not feature cutoff for the MaxEnt 3-gram model. Next, we observe that the supervised adaptation algorithm `Prior` yields significant reductions in WER, from 25.7% to 22.8%. This suggests that the MIT domain differs from the BN domain and access to transcribed text can significantly improve the model.

Turning towards the more challenging semi-supervised setting, our CE Regularization method improves both over the out-of-domain baselines (25.7 to 24.9) but also the semi-supervised baseline `Self-Train`. Furthermore, our method recovers 28% of the possible improvement between the baseline and fully supervised method, as compared to 14% for `Self-Train`. It is interesting that both semi-supervised methods result in a similar perplexity, while our model still gives a further WER reduction, possibly due to the discriminative nature of our method. Finally, we can observe that conditional entropy minimization is a desirable property of a language model. Table III shows the mean conditional entropy of each language model on the adaptation data. As expected, since CE regularization is explicitly minimizing this quantity it obtains lower CE than the other semi-supervised and unadapted methods. However, the supervised baseline obtains an even lower mean CE, suggesting that further reductions may be associated with improved model performance.

VII. CONCLUSION

We have presented a new language model training objective for model adaptation based on conditional entropy minimization. Additionally, we provide an algorithm for efficiently computing the conditional entropy of speech lattices using expectation semirings. By training our model on out-of-domain text data and regularizing on in-domain audio, we achieve larger reductions in WER than other popular semi-supervised adaptation techniques. Additionally, we demonstrate progress in closing the gap between semi-supervised and supervised adaptation algorithms.

VIII. ACKNOWLEDGEMENTS

This research was partially supported by National Science Foundation Grant No 0963898 and by the JHU Human Language Technology Center of Excellence. Bhuvana Ramabhadran, Abhinav Sethy and Brian Kingsbury from IBM provided valuable assistance, and the acoustic models used in the experiments whose results are reported here.

REFERENCES

[1] R. Rosenfeld, "A maximum entropy approach to adaptive statistical language modelling," *Computer Speech and Language*, vol. 10, no. 3, pp. 187–228, 1996.
[2] T. Alumäe and M. Kurimo, "Efficient Estimation of Maximum Entropy Language Models with N -gram features : an SRILM extension," in *Interspeech*, 2010.
[3] J. Wu, "Maximum Entropy Language Modeling with Non-Local Dependencies," Ph.D. dissertation, Johns Hopkins University, 2002.
[4] Y. Grandvalet and Y. Bengio, "Semi-supervised learning by entropy minimization," in *NIPS*, vol. 17, 2004, pp. 529–536.
[5] F. Jiao, S. Wang, C. Lee, R. Greiner, and D. Schuurmans, "Semi-supervised conditional random fields for improved sequence segmentation and labeling," in *COLING/ACL*, 2006.
[6] H. Li, K. Zhang, and T. Jiang, "Minimum entropy clustering and applications to gene expression analysis," in *IEEE Computational Systems Bioinformatics Conference*, 2004, pp. 142–151.
[7] A. Rastrow, F. Jelinek, A. Sethy, and B. Ramabhadran, "Unsupervised Model Adaptation using Information-Theoretic Criterion," in *NAACL*, 2010.
[8] Z. Li and J. Eisner, "First- and second-order expectation semirings with applications to minimum-risk training on translation forests," in *EMNLP*, 2009.
[9] C. Chelba and A. Acero, "Adaptation of maximum entropy capitalizer: Little data can help a lot," *Computer Speech & Language*, 2006.
[10] M. Bacchiani and B. Roark, "Unsupervised language model adaptation," in *ICASSP*, 2003, pp. 224–227.
[11] S. Della Pietra, V. Della Pietra, and J. Lafferty, "Inducing features of random fields," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 19, no. 4, pp. 380–393, 1997.
[12] D. C. Liu and J. Nocedal, "On the limited memory BFGS method for large scale optimization methods," *Mathematical Programming*, vol. 45, pp. 503–528, 1989.
[13] S. Chen and R. Rosenfeld, "A Gaussian prior for smoothing maximum entropy models," *Technical Report*, 1999.
[14] T. M. Cover and J. A. Thomas, *Elements of information theory*, 3rd ed. Wiley-Interscience, 2006.
[15] S. Chen, B. Kingsbury, L. Mangu, D. Povey, G. Saon, H. Soltau, and G. Zweig, "Advances in speech transcription at IBM under the DARPA EARS program," *IEEE Transactions on Audio, Speech and Language Processing*, pp. 1596–1608, 2006.
[16] J. Glass, T. Hazen, S. Cyphers, I. Malioutov, D. Huynh, and R. Barzilay, "Recent progress in MIT spoken lecture processing project," in *Interspeech*, 2007.