MODEL COMBINATION FOR SPEECH RECOGNITION USING EMPIRICAL BAYES RISK MINIMIZATION

Anoop Deoras†, Denis Filimonov§‡, Mary Harper§‡and Fred Jelinek†‡

†Center for Language and Speech Processing,
‡Human Language Technology Center of Excellence,
Johns Hopkins University, Baltimore MD 21218, USA
§UMIACS Laboratory for Computational Linguistics and Information Processing
University of Maryland, College Park MD 20742, USA
adeoras@jhu.edu, den@cs.umd.edu, mharper@umd.edu, jelinek@jhu.edu

ABSTRACT

In this paper, we explore the model combination problem for rescoring Automatic Speech Recognition (ASR) hypotheses. We use minimum Empirical Bayes Risk for the optimization criterion and Deterministic Annealing techniques to search through the non-convex parameter space. Our experiments on the DARPA WSJ task using several different language models showed that our approach consistently outperforms the standard methods of model combination that optimize using 1-*best* hypothesis error.

Index Terms— Discriminative Model Combination, Deterministic Annealing

1. INTRODUCTION

The conventional Maximum a Posteriori (MAP) approach [1] to ASR aims to find the most likely word sequence $\mathbf{W}^*$ given the acoustic observation $\mathbf{A}$, which is typically represented in the following way:

$$\mathbf{W}^* = \arg\max_W P(\mathbf{W}|\mathbf{A}) = \arg\max_W P(\mathbf{A}|\mathbf{W})P(\mathbf{W})$$

where $P(\mathbf{A}|\mathbf{W})$ is the characterization of an Acoustic Model (AM) and $P(\mathbf{W})$ is the characterization of a Language Model (LM). The probability of the word sequence $W \equiv w_1, \ldots, w_m \equiv w_1^m$ is typically factored using the chain rule:

$$P(w_1^m) = \prod_{i=1}^{m} P(w_i|w_1^{i-1}) \quad (1)$$

In modern recognition systems, an LM tends to be restricted to simple n-gram models, where the distribution of the predicted word depends on the previous $(n-1)$ words i.e. $P(w_i|w_1^{i-1}) \approx P(w_i|w_{i-n+1}^{i-1})$. Due to the complexity of the search involved in large vocabulary speech recognition systems, incorporating a higher order LM increases the memory requirements, as well as the time complexity of the decoder [2]. Due to this limitation, the order of the LM used for recognition is typically restricted to 3 or 4. At the same time, number of n-grams, i.e., $\{w_i, w_{i-n+1}^{i-1}\}$ pairs, cannot total more than a few million (typically 5 million is the limit observed in our systems). However, pruning the LM (in order and size) results in a lower performance of the recognition system. To accommodate to this situation, typically, instead of a one best hypothesis, many hypotheses are output by the recognizers and then a more complex LM can be used for rescoring.

Recognition hypotheses can be compactly represented in a data structure called a *lattice*, in which each edge represents a word. A path from the source of the lattice to its sink corresponds to a hypothesis. Each edge is annotated with probabilities to generate the word and the pronunciation given the state represented at the edge's source node, and the shortest path algorithm (Viterbi decoding) can be used to extract the best hypothesis. However, in order to support re-scoring with more complex language models [3, 4, 5] (e.g., higher order n-gram models, syntactic language models), more information must be associated with the states in the lattice, which can make re-scoring intractable[1]. Therefore, other approaches are needed, such as a greedy search in a lattice (or a confusion network as in [6]) or N best hypothesis extraction (based on the weaker model) followed by re-scoring with the more sophisticated LM.

Regardless of which representation for the space of hypotheses we choose, the re-scoring paradigm offers an opportunity to apply a combination of of several complex language models, which (provided the models are sufficiently complementary) should be able to outperform any single model in the set. This raises the question of how to combine the models in order to achieve maximum performance from the model combination.

[1]In addition there are classes of language models that assign scores to entire sentences rather than using Eq. 1. Scores for these models simply cannot be represented in a lattice.

Peter Beyerlein [7] optimized his model combinations for minimum 1-best error. However this objective function is piecewise constant, prohibiting the use of Gradient-Descent-like optimization methods. In his work, he smoothed the 1-*best* loss using various squashing functions. After seeing promising results in using the Empirical Bayes Risk objective function for *decoding*, Goel [8] proposed using this objective function for *training* speech recognition systems in the re-scoring framework as a natural extension to Beyerlein's work. Smith et.al. [9] later demonstrated for a Machine Translation (MT) and Dependency Parsing task, that changing the objective function from 1-*best* loss to expected loss (i.e. Bayes Risk) has the added advantage of making it a smoother function of model parameters, enabling the use of sophisticated Gradient-Descent-based optimization techniques.

In this work, we focus on the log-linear combination (detailed in Section 2) of several different models, using, in particular, the Deterministic Annealing framework (described in Section 4) to carry out non-convex optimization based on the Empirical Bayes Risk objective function (described in Section 3). To the best of our knowledge, there has been no prior work on model combination using the Empirical Bayes Risk objective function for speech recognition systems. Although Rao et.al.[10] trained an *isolated* speech recognition system using Deterministic Annealing (DA) to minimize the smoothed error function, it is the goal of this paper to demonstrate its efficacy when re-scoring a Large Vocabulary Speech Recognition (LVCSR) task using combinations of various n-gram and syntactic language models. In Section 5, we present our experimental setup and results. We analyze our findings in Section 6 and conclude the paper and outline future work in Section 7.

2. LOG LINEAR MODEL

We combine various recognition models into one unified log linear model. We use scores or probabilities assigned to entire hypotheses, thus our framework can accommodate whole-sentence discriminative models, as well as generative models factorizable as shown in Eq. 1.

Let us establish some notation. Let us denote our dataset by T, and let there be $|\mathcal{T}|$ speech utterances to decode and to rescore the recognizer's corresponding first pass search space. We will index these speech utterances by the letter $\mathbf{o}_t$, where $t = \{1, 2, \ldots, |\mathcal{T}|\}$. Let the search space output by the recognizer be N best lists. Thus for each speech utterance $\mathbf{o}_t$, we will have top N best hypotheses according to some baseline model. Let these N best hypotheses[2] be denoted by $W_{t,1}, \ldots, W_{t,N}$. For each speech utterance, $\mathbf{o}_t$, let us denote the corresponding true transcript by $W_{t,0}$.

Let there be M models, each assigning scores to the N hypotheses. Let us denote these models by $q_m(\cdot|\mathbf{o}_\cdot)$, where $m = \{1, 2, \ldots, M\}$. Thus under any m^{th} model, the score of n^{th} hypothesis, $W_{t,n}$, obtained after the first pass decoding of t^{th} speech utterance, will be denoted as $q_m(W_{t,n}|\mathbf{o}_t)$. We now define our log linear model $P_\Lambda(\cdot|\mathbf{o}_\cdot)$ as shown below:

$$P_\Lambda(W_{t,n}|\mathbf{o}_t) = \frac{(\exp\{\sum_{m=1}^{M} \lambda_m q_m(W_{t,n}|\mathbf{o}_t)\})}{\sum_{k=1}^{N} \exp\{\sum_{m=1}^{M} \lambda_m q_m(W_{t,k}|\mathbf{o}_t)\})} \quad (2)$$

where λ_m is the model weight for m^{th} recognition model.

3. OBJECTIVE FUNCTION

Under a loss function, viz. word edit distance, $\mathcal{L}(\cdot,\cdot)$, we can define the Expected-Word Error Rate (E-WER) under some model, $P_{\Lambda,\gamma}(\cdot)$, as:

$$\mathbf{E}_{P_\Lambda}[\mathcal{L}] = \frac{1}{|\mathcal{R}|} \sum_{t=1}^{|\mathcal{T}|} \sum_{n=1}^{N} \mathcal{L}(W_{t,n}, W_{t,0}) P_\Lambda(W_{t,n}|\mathbf{o}_t) \quad (3)$$

where, $|\mathcal{R}|$ is the sum of the length of $|\mathcal{T}|$ true transcripts. We can thus rewrite the above Expected loss function as:

$$\begin{aligned} \mathbf{E}_{P_\Lambda}[\mathcal{L}] &= \frac{1}{|\mathcal{T}|} \sum_{t=1}^{|\mathcal{T}|} \sum_{n=1}^{N} \frac{|\mathcal{T}|}{|\mathcal{R}|} \mathcal{L}(W_{t,n}, W_{t,0}) P_\Lambda(W_{t,n}|\mathbf{o}_t) \\ &= \frac{1}{|\mathcal{T}|} \sum_{t=1}^{|\mathcal{T}|} \sum_{n=1}^{N} \mathcal{L}'(W_{t,n}, W_{t,0}) P_\Lambda(W_{t,n}|\mathbf{o}_t) \end{aligned}$$

where $\mathcal{L}'(\cdot,\cdot) = \frac{|\mathcal{T}|}{|\mathcal{R}|}\mathcal{L}(\cdot,\cdot)$. Henceforth, we will use $\mathcal{L}(\cdot,\cdot)$ to mean $\mathcal{L}'(\cdot,\cdot)$.

4. DETERMINISTICALLY ANNEALED TRAINING

For a non-convex objective function, use of any hill climbing method results in the locally optimum solution, which may or may not be the globally optimum solution. For our case, unfortunately, E-WER is a non-convex function of the model parameters and hence use of Gradient-Descent-based optimization method does not guarantee finding the globally optimum point. Deterministic Annealing [10] attempts to solve this problem by introducing a hyper parameter γ to Eq. 2, which modifies it as shown below:

$$P_{\Lambda,\gamma}(W_{t,n}|\mathbf{o}_t) = \frac{(\exp\{\sum_{m=1}^{M} \lambda_m q_m(W_{t,n}|\mathbf{o}_t)\})^\gamma}{\sum_{k=1}^{N} \exp\{\sum_{m=1}^{M} \lambda_m q_m(W_{t,k}|\mathbf{o}_t)\})^\gamma} \quad (4)$$

When the value of γ is set to 0, the log linear model assigns uniform distribution to all the hypotheses, resulting in a higher entropy model. When γ is set to value 1, then the log linear model takes the form of Eq. 2 and when $\gamma \to \infty$, the model approaches the form where all probability mass is concentrated towards one specific hypothesis, resulting in a much

[2]For simplicity, we assume that N is the same for all speech utterances, although it is not required.

lower entropy distribution. If the parameters of the model are trained to minimize 1-*best* loss, then the distribution mass is concentrated towards the best hypothesis.

As very neatly analyzed in [9], for a fixed γ, deterministic annealing solves:

$$\begin{aligned}\Lambda^* &= \arg\min_{\Lambda} \mathbf{E}_{P_{\Lambda,\gamma}}[\mathcal{L}(\cdot,\cdot)] \\ &= \arg\min_{\Lambda} \frac{1}{|\mathcal{T}|}\sum_{t=1}^{|\mathcal{T}|}\sum_{n=1}^{N}\mathcal{L}(W_{t,n},W_{t,0})P_{\Lambda,\gamma}(W_{t,n}|\mathbf{o}_t)\end{aligned}$$

When γ is increased according to some schedule, Λ is optimized again. Since lower values of γ smooths the distribution, this may allow us to avoid the locally optimum solution, which otherwise may not be possible under relatively higher values of γ.[3]

Given the form of the model in Eq. 4, γ can be multiplied by Λ and taken inside the exponentiation. Hence any change in γ can be easily compensated for a respective change in the value of Λ vector. Hence in [10] a direct preference for the Shannon entropy, H, was expressed instead. Thus, γ and Λ are chosen according to:

$$\begin{aligned}\{\Lambda^*,\gamma^*\} &= \arg\min_{\Lambda,\gamma} \mathbf{E}_{P_{\Lambda,\gamma}}[\mathcal{L}(\cdot,\cdot)] - \Theta\mathrm{H}(P_{\Lambda,\gamma}(\cdot)) \\ &= \arg\min_{\Lambda,\gamma} \frac{1}{|\mathcal{T}|}\sum_{t=1}^{|\mathcal{T}|}\sum_{n=1}^{N}\mathcal{L}(W_{t,n},W_{t,0})P_{\Lambda,\gamma}(W_{t,n}|\mathbf{o}_t) \\ &\quad +\Theta\frac{1}{|\mathcal{T}|}\sum_{t=1}^{|\mathcal{T}|}\sum_{n=1}^{N}P_{\Lambda,\gamma}(W_{t,n}|\mathbf{o}_t)\log P_{\Lambda,\gamma}(W_{t,n}|\mathbf{o}_t) \\ &= \arg\min_{\Lambda,\gamma} \frac{1}{|\mathcal{T}|}\sum_{t=1}^{|\mathcal{T}|}\sum_{n=1}^{N}P_{\Lambda,\gamma}(W_{t,n}|\mathbf{o}_t)\times \\ &\quad \Big(\mathcal{L}(W_{t,n},W_{t,0}) + \Theta\log P_{\Lambda,\gamma}(W_{t,n}|\mathbf{o}_t)\Big) \end{aligned} \quad (5)$$

where, Θ is the temperature. Let us denote the above objective function by $F(\Lambda,\gamma,\Theta)$. We seek to find the minimum of $F(\cdot)$ and the corresponding parameters achieving this minima. Algorithm 1 illustrates Deterministic-Annealing-based model search.

[3]The assumption here is that the locally optimum solutions do not have a very peaky descent and hence smoothing the distribution smoothes away any locally optimum descent and sharpens the globally optimum descent valley relative to all the others.

Input: $\mathbf{W}$, $\mathbf{q}$, $\Theta_{start} > \Theta_{final}$, H_{min}, γ_{start}, γ_{final}
Output: $\Lambda^*_{\gamma^*}, \gamma^*$

$\Theta = \Theta_{start}$, $\gamma = \gamma_{start}$

# Cooling i.e. decrease Θ with fixed γ
while $\Theta > \Theta_{final}$ **do**
  $\Lambda_{\Theta,\gamma} = \arg\min_{\Lambda} \mathbf{E}_{P_{\Lambda,\gamma}}[\mathcal{L}(\cdot,\cdot)] - \Theta\mathrm{H}(P_{\Lambda,\gamma})$
  $\Theta = \alpha(\Theta)$
end
# Quenching i.e. increase γ with fixed $\Theta = 0$
while $\gamma < \gamma_{final}$ && $H > H_{min}$ **do**
  $\Lambda_{\Theta,\gamma} = \arg\min_{\Lambda} \mathbf{E}_{P_{\Lambda,\gamma}}[\mathcal{L}(\cdot,\cdot)]$;
  $\gamma = \beta(\gamma)$
end

Algorithm 1: Algorithm for Deterministic Annealing

For both cooling and quenching steps, we need to find the model parameters that minimize the respective objective functions. This is achieved by a Gradient-Descent-based optimization method. The model parameters obtained at the end of any particular cooling schedule becomes the initialization for the optimization of the next cooling step (similarly for quenching steps). Thus each time the Gradient-Descent-based optimization method is invoked, the objective function takes a different form based on the parameters and hyper-parameters learned from previous steps.

4.1. Partial Derivatives

To carry out Gradient-Descent-based optimization, we need partial derivatives of the objective function with respect to the model weights. The partial derivative of either $F(\cdot)$ or $E[\cdot](\mathcal{L}(\cdot,\cdot))$ requires the partial derivative of $P_{\Lambda,\gamma}(W_{t,n}|\mathbf{o}_t)$. The partial derivative of $F(\cdot,\cdot,\cdot)$ with respect to some m^{th} model parameter λ_m is given by:

$$\begin{aligned}\frac{\partial F(\Lambda,\gamma,\Theta)}{\partial\lambda_m} &= \frac{1}{|\mathcal{T}|}\sum_{t=1}^{|\mathcal{T}|}\sum_{n=1}^{N}\mathcal{L}(W_{t,n},W_{t,0})\frac{\partial P_{\Lambda,\gamma}(W_{t,n}|\mathbf{o}_t)}{\partial\lambda_m} \\ &\quad +\Theta\frac{1}{|\mathcal{T}|}\sum_{t=1}^{|\mathcal{T}|}\sum_{n=1}^{N}\frac{\partial P_{\Lambda,\gamma}(W_{t,n}|\mathbf{o}_t)\log P_{\Lambda,\gamma}(W_{t,n}|\mathbf{o}_t)}{\partial\lambda_m} \\ &= \frac{1}{|\mathcal{T}|}\sum_{t=1}^{|\mathcal{T}|}\sum_{n=1}^{N}\frac{\partial P_{\Lambda,\gamma}(W_{t,n}|\mathbf{o}_t)}{\partial\lambda_m}\Big(\mathcal{L}(W_{t,n},W_{t,0}) \\ &\quad +\Theta(1+\log P_{\Lambda,\gamma}(W_{t,n}|\mathbf{o}_t))\Big)\end{aligned}$$

Similarly, the partial derivative of the Expected loss is given by:

$$\frac{\partial\mathbf{E}_{P_{\Lambda,\gamma}}[\mathcal{L}]}{\partial\lambda_m} = \frac{1}{|\mathcal{T}|}\sum_{t=1}^{|\mathcal{T}|}\sum_{n=1}^{N}\mathcal{L}(W_{t,n},W_{t,0})\frac{\partial P_{\Lambda,\gamma}(W_{t,n}|\mathbf{o}_t)}{\partial\lambda_m} \quad (6)$$

We thus need to determine $\frac{\partial P_{\Lambda,\gamma}(W_{t,n}|\mathbf{o}_t)}{\partial \lambda_m}$ in order to solve the above equations.

Before we obtain the above partial derivative, let us establish some notation. Let us represent the numerator term of Eq. 4, i.e. the un-normalized probability, by $\tilde{P}_{\Lambda,\gamma}(W_{t,n}|\mathbf{o}_t)$. Let the denominator be denoted by $Z(t)$. Note that $Z(t)$ is simply the sum of un-normalized probabilities over all the N best hypotheses for a particular speech utterance $\mathbf{o}_t$. Each speech utterance, $\mathbf{o}_t$, will have its corresponding term $Z(t)$. Hence, we can rewrite the log linear equation as:

$$\begin{aligned} P_{\Lambda,\gamma}(W_{t,n}|\mathbf{o}_t) &= \frac{\tilde{P}_{\Lambda,\gamma}(W_{t,n}|\mathbf{o}_t)}{Z(t)} \\ &= \frac{\tilde{P}_{\Lambda,\gamma}(W_{t,n}|\mathbf{o}_t)}{\sum_{k=1}^{N}\tilde{P}_{\Lambda,\gamma}(W_{t,k}|\mathbf{o}_t)} \end{aligned}$$

The partial derivative of $P(\cdot)$ can be obtained as:

$$\frac{\partial P(\cdot)}{\partial \lambda_m} = \frac{\partial P(\cdot)}{\partial \log P(\cdot)}\frac{\partial \log P(\cdot)}{\partial \lambda_m}$$

Note, the first term on the right hand side is simply $P(\cdot)$. The second term can be obtained as shown below:

$$\begin{aligned} &\frac{\partial \log P_{\Lambda,\gamma}(W_{t,n}|\mathbf{o}_t)}{\partial \lambda_m} = \\ &= \frac{\partial \log \tilde{P}_{\Lambda,\gamma}(W_{t,n}|\mathbf{o}_t)}{\partial \lambda_m} - \frac{\partial \log Z(t)}{\partial \lambda_m} \\ &= \frac{\partial(\sum_{m'=1}^{M}\lambda'_m \gamma q'_m(W_{t,n}|\mathbf{o}_t)))}{\partial \lambda_m} - \frac{\partial \log Z(t)}{\partial \lambda_m} \\ &= \gamma q_m(W_{t,n}|\mathbf{o}_t))) - \frac{1}{Z(t)}\sum_{k=1}^{N}\tilde{P}_{\Lambda,\gamma}(W_{t,k}|\mathbf{o}_t)\gamma q_m(W_{t,k}) \end{aligned}$$

Thus the partial derivative of $P(\cdot)$ can be obtained as:

$$\begin{aligned} \frac{\partial P_{\Lambda,\gamma}(W_{t,n}|\mathbf{o}_t)}{\partial \lambda_m} &= \gamma P_{\Lambda,\gamma}(W_{t,n}|\mathbf{o}_t)\Big(q_m(W_{t,n}) \\ &\quad - \frac{1}{Z(t)}\sum_{k=1}^{N}\tilde{P}_{\Lambda,\gamma}(W_{t,k}|\mathbf{o}_t)q_m(W_{t,k})\Big) \end{aligned}$$

5. EXPERIMENTS AND RESULTS

We evaluated combinations of various models with various optimization methods (suited for the corresponding objective function). We used 100-best lists from DARPA WSJ'92 and WSJ'93 20k open vocabulary data sets. The acoustic model and baseline bi-gram LM, used to generate the N-best list can be found in [11]. We used the 93et and 93dt sets for development (i.e. for tuning various parameters using various optimization methods) and 92et for evaluation. The development set contained a total of 465 utterances, while the evaluation set contained a total of 333 utterances.

We trained a Kneser-Ney smoothed tri-gram LM built from $70M$ words of the NYT section of the English Gigaword. We used the $20k$ vocabulary supplied with the WSJ'93 dataset. The vocabulary, as well as the N-best lists, were tokenized to match the Penn Treebank style, namely contractions and possessives were separated. The baseline bigram LM and the tri-gram LM are the n-gram LMs.

We also obtained LM scores for N-best lists using syntactic LMs (Filimonov et.al.'s [5]) which use different tagsets to represent syntactic information, namely *head* (which captures dependency), *parent* (constituency), and *SuperARV* (dependency and lexical features). The syntactic models have been trained on data produced from the same NYT section parsed automatically. We refer the reader to [5] for details on the training procedure and the syntactic models.

In all, we used 6 statistical models viz. baseline acoustic model (am), baseline bi-gram language model (bg), rescoring trigram language model (tg), and three syntactic language models using different tagsets: *head*, *parent*, and *SuperARV* (sarv).

We evaluated model combination using following model parameter search methods:

1. **Exhaustive Grid Search:** Exhaustive grid search was performed to find model weights when the various submodels are combined as in Eq. 2.

2. **MERT:** Powell's line search [12] was performed to carry out Minimum Error Rate Training (MERT), i.e., to optimize non-smooth 1-*best* error to find model weights when the models are combined as shown in Eq. 2. We made use of SRILM toolkit's [13] *nbest-optimize* functionality. We constrained the search to non-negative model weights. We tried many random initializations and chose the one which gave better performance on the development dataset.

3. **Minimum Risk:** Our proposed model search was performed to optimize smoothed E-WER to find model weights when the models are combined as shown in Eq. 2. Deterministic Annealing was used in conjunction with minimum risk training. We use L-BFGS[4] to solve all the Gradient-Descent-based optimization problems. Discussion of the cooling and quenching schedule of Deterministic Annealing appears in section 4.

In our work, we focus on comparing the performance of the 1-*best* error minimization objective function to the expected word error minimization optimization function. It would have been fairer to smooth the 1-best loss with a squashing function like sigmoid. This would have enabled

[4]Broyden-Fletcher-Goldfarb-Shanno (BFGS) method is an effective, often used method for solving nonlinear optimization problems. L-BFGS [14] is the low memory implementation of BFGS.

the use of Gradient-Descent-based optimization methods for both objective functions. In this way, the analysis of the two methods would have directly compared 1-best loss performance to expected loss, rather than comparing a non-smooth objective function search method to a smooth objective function search method. However, there is no single best method for smoothing the non-smooth 1-*best* loss function. Powell's line search, on the other hand, is a very well known optimization algorithm for non-smooth functions. Years of engineering has made the implementation of this algorithm very robust in state of the art LM toolkits such as the SRILM toolkit. We therefore avoided smoothing the 1-best loss objective function in this work, although, we plan to investigate it in our future work. In this work, we instead performed many random restarts and picked the parameter that minimized the objective function the greatest on the training set.

5.1. Implementation details

For minimizing the risk, we allowed Temperature, Θ to range from 1 to 0 in steps of 0.1. We allowed γ to range from 0.1 to 2, again in steps of 0.1. We noticed that quenching did not have any significant effect and it were only the cooling steps that helped. Moreover, Deterministic Annealing based technique outperformed vanilla Gradient Descent based technique only when the number of model increased beyond a certain point.

Setup	Dev	Eval
am+bg (Exh-Search)	18.2	13.0
am+bg (MERT)	18.2	13.1
am+bg (MinRisk)	18.2	13.0
am+tg (Exh-Search)	17.4	12.2
am+tg (MERT)	17.4	12.4
am+tg (MinRisk)	17.4	12.2
am+head (Exh-Search)	16.8	11.8
am+head (MERT)	16.9	11.9
am+head (MinRisk)	16.8	11.8
am+parent (Exh-Search)	16.9	11.7
am+parent (MERT)	16.9	11.7
am+parent (MinRisk)	16.9	11.7
am+sarv (Exh-Search)	16.9	11.8
am+sarv (MERT)	16.9	11.8
am+sarv (MinRisk)	16.9	11.8

Table 1. Results (Word Error Rates (in %)) for combinations of two models.

6. DISCUSSION

From Table 1, we observe that the performance of both the 1-*best* loss and expected loss, objective function is comparable in almost all the model combinations. Moreover, the performance matches that obtained with exhaustive model parameter search. When only two models are combined, then typically by keeping the weight of one model fixed to 1, the weight of the other model can be appropriately renormalized. This normalization does not affect MAP decoding. Carrying out exhaustive search thus boils down to searching exhaustively in just one dimension by fixing the other dimension to the value of 1[5].

Setup	Dev	Eval
am+bg+tg (MERT)	16.5	11.7
am+bg+tg (MinRisk)	**16.0**	**10.9**
am+bg+tg+head (MERT)	15.9	11.2
am+bg+tg+head (MinRisk)	16.0	11.0
am+bg+tg+parent (MERT)	16.1	11.2
am+bg+tg+parent (MinRisk)	16.1	**10.9**
am+bg+tg+sarv (MERT)	16.0	11.4
am+bg+tg+sarv (MinRisk)	16.0	**11.1**
am+bg+tg+head+parent (MERT)	16.0	10.9
am+bg+tg+head+parent (MinRisk)	15.8	**10.4**
am+bg+tg+head+parent+sarv (MERT)	16.0	11.1
am+bg+tg+head+parent+sarv (MinRisk)	**15.6**	**10.5**

Table 2. Results (Word Error Rates (in %)) for combinations of three or more models.

When the number of models used in combination increases beyond two, then the search has to be carried out exhaustively in two or more dimensions. Although this could done, it would be prohibitively slow, and so we prefer to avoid exhaustive search. In the case of two model combinations, we observed that the objective function was nearly convex with respect to the model parameters; hence, a hill climbing method indeed finds a globally optimum solution. However, when the number of models increases beyond two, the objective function becomes more and more non-convex. From Table 2, we can see that the model search using 1-best loss as the objective function fails to find the globally optimum solution. However, using E-WER as the objective function, regularized with the entropy, we obtain a much better model that generalizes well even on the evaluation data set. Our proposed method consistently performs better than MERT. In the case where we combine all the statistical models (last row of Table 2), using our proposed approach, we obtain about a $\mathbf{0.6}\%$ absolute WER improvements when compared to the 1-*best* loss objective function based model search.

It is important to note that we reconfirm Beyerlein's observation [7] that by combining complementary models in a unified framework, one can achieve much better recognition performance than the best performance obtained by any of the sub-models. In our work when we combine all the models to-

[5]However it should be noted that for MBR [8] the normalization affects the decision boundaries during decoding. Hence even for a simple case of two model combination, the search for parameters needs to be carried out exhaustively in two dimensions.

gether (last row of Table 2), we obtain a $\mathbf{1.3}\%$ absolute WER decrease from the WER obtained using the best sub-model (i.e. the syntactic LM using the tag *parent*).

In conclusion, when we have complementary knowledge sources, by way of model combination, it is possible to achieve improved recognition performance. By having a smooth objective function like expected loss (E-WER in our case), the model search can be done in a more principled manner resulting in a much better, generalized model.

7. FUTURE DIRECTIONS

In future work, we will investigate smoothing the loss function using the sigmoid or other similar squashing functions and then carry out the Gradient-Descent based optimization method along with Deterministic Annealing. This would enable us to directly compare the performance of 1-*best* loss objective function to the expected loss objective function with the same search algorithm.

We also plan to carry out model training and decoding using combinations of models on word lattices rather than N best lists. It should be noted, however, that in this work, we have used some models that require the complete word string hypothesis in order to assign the probability of the word given its past history. Re-scoring word lattices using these complex models becomes intractable. As a way to avoid this problem, we plan to use Deoras et.al.'s Iterative Decoding framework [6, 15] to carry out re-scoring. We also plan to adapt the algorithm so that it can be used during both training and decoding.

Finally, we will also investigate why some models combine to produce better results than others. For example, we could evaluate various principled methods to create more effective model combinations.

8. ACKNOWLEDGEMENT

The authors would like to thank the anonymous reviewers for their comments. This research was supported by NSF OISE-0530118 and in part by NSF IIS-0703859. Any opinions, findings, and recommendations expressed in this paper are those of the authors and do not necessarily reflect the views of the funding agency or the institution where the work was completed.

9. REFERENCES

[1] L. R. Bahl, F. Jelinek, and R. L. Mercer, "A maximum likelihood approach to continuous speech recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1983.

[2] Frederick Jelinek, *Statistical methods for speech recognition*, MIT Press, Cambridge, MA, USA, 1997.

[3] Roni Rosenfeld, "A maximum entropy approach to adaptive statistical language modelling," *Computer Speech and Language*, vol. 10, pp. 187–228, 1996.

[4] Hong-Kwang Jeff Kuo et.al., "Morphology for arabic speech recognition," in *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 2010.

[5] Denis Filimonov and Mary Harper, "A joint language model with fine-grain syntactic tags," in *Proc of 2009 Conference on Empirical Methods in Natural Language Processing*, 2009.

[6] Anoop Deoras and Frederick Jelinek, "Iterative decoding: A novel re-scoring framework for confusion networks," in *Proc. IEEE Automatic Speech Recognition and Understanding (ASRU)*, 2009.

[7] Peter Beyerlein, "Discriminative model combination," in *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 1998.

[8] Vaibhava Goel, *Minimum Bayes Risk Automatic Speech Recogntion*, Ph.D. thesis, The Johns Hopkins University, 2000, Adviser-Byrne, W.

[9] David Smith and Jason Eisner, "Minimum risk annealing for training log-linear models," in *Proc of the COLING/ACL 2006*, 2006.

[10] Ajit V. Rao and Kenneth Rose, "Deterministically annealed design of hidden markov model speech recognizers," *IEEE Transactions on Speech and Audio Processing*, vol. 9, 2001.

[11] Wen Wang and Mary Harper, "The superarv language model: investigating the effectiveness of tightly integrating multiple knowledge sources," in *Proc of the EMNLP 2002*, 2002.

[12] W. H. Press, B. P. Flannery, S. A. Teukolsky, and W. T. Vetterling, *Numerical Recipes in C: The Art of Scientific Computing*, Cambridge University Press, 1988.

[13] A. Stolcke, "Srilm – an extensible language modeling toolkit," in *Proc. Intl. Conf. on Spoken Language Processing (ICSLP), 2002.*, 2002.

[14] D.C. Liu and J. Nocedal, *On the Limited Memory Method for Large Scale Optimization (Mathematical Programming)*, 1989.

[15] Anoop Deoras, "Simulated annealing based extensions to iterative decoding," *Quarterly Technical Report, HLT-COE, Johns Hopkins University*, 2010.