Regularization and Search for Minimum Error Rate Training

Daniel Cer, **Daniel Jurafsky**, and **Christopher D. Manning**
Stanford University
Stanford, CA 94305
cerd,jurafsky,manning@stanford.edu

Abstract

Minimum error rate training (MERT) is a widely used learning procedure for statistical machine translation models. We contrast three search strategies for MERT: Powell's method, the variant of coordinate descent found in the Moses MERT utility, and a novel stochastic method. It is shown that the stochastic method obtains test set gains of +0.98 BLEU on MT03 and +0.61 BLEU on MT05. We also present a method for regularizing the MERT objective that achieves statistically significant gains when combined with both Powell's method and coordinate descent.

1 Introduction

Och (2003) introduced minimum error rate training (MERT) as an alternative training regime to the conditional likelihood objective previously used with log-linear translation models (Och & Ney, 2002). This approach attempts to improve translation quality by optimizing an automatic translation evaluation metric, such as the BLEU score (Papineni et al., 2002). This is accomplished by either directly walking the error surface provided by an evaluation metric w.r.t. the model weights or by using gradient-based techniques on a continuous approximation of such a surface. While the former is piecewise constant and thus cannot be optimized using gradient techniques, Och (2003) provides an approach that performs such training efficiently.

In this paper we explore a number of variations on MERT. First, it is shown that performance gains can be had by making use of a stochastic search strategy as compare to that obtained by Powell's method and coordinate descent. Subsequently, results are presented for two regularization strategies[1]. Both allow coordinate descent and Powell's method to achieve performance that is on par with stochastic search.

In what follows, we briefly review minimum error rate training, introduce our stochastic search and regularization strategies, and then present experimental results.

2 Minimum Error Rate Training

Let $\mathbf{F}$ be a collection of foreign sentences to be translated, with individual sentences $\mathbf{f}_0$, $\mathbf{f}_1$, $\ldots$, $\mathbf{f}_n$. For each $\mathbf{f}_i$, the surface form of an individual candidate translation is given by $\mathbf{e}_i$ with hidden state $\mathbf{h}_i$ associated with the derivation of $\mathbf{e}_i$ from $\mathbf{f}_i$. Each $\mathbf{e}_i$ is drawn from $\mathcal{E}$, which represents all possible strings our translation system can produce. The $(\mathbf{e}_i, \mathbf{h}_i, \mathbf{f}_i)$ triples are converted into vectors of m feature functions by $\Psi : \mathcal{E} \times \mathcal{H} \times \mathcal{F} \rightarrow \mathbb{R}^m$ whose dot product with the weight vector $\mathbf{w}$ assigns a score to each triple. The idealized translation process then is to find the highest scoring pair $(\mathbf{e}_i, \mathbf{h}_i)$ for each $\mathbf{f}_i$, or rather $(\mathbf{e}_i, \mathbf{h}_i) = \mathrm{argmax}_{(\mathbf{e} \in \mathcal{E}, \mathbf{h} \in \mathcal{H})} \mathbf{w} \cdot \Psi(\mathbf{e}, \mathbf{h}, \mathbf{f})$.

The aggregate argmax for the entire data set $\mathbf{F}$ is given by equation (1)[2]. This gives $\mathbf{E_w}$ which represents the set of translations selected by the model for data set $\mathbf{F}$ when parameterized by the weight vector $\mathbf{w}$. Let's assume we have an automated measure of translation quality ℓ that maps the collec-

[1] While we prefer the term regularization, the strategies presented here could also be referred to as smoothing methods.

[2] Here, the translation of the entire data set is treated as a single structured prediction problem using the feature function vector $\Psi(\mathbf{E}, \mathbf{H}, \mathbf{F}) = \sum_i^n \Psi(\mathbf{e}_i, \mathbf{h}_i, \mathbf{f}_i)$

Proceedings of the Third Workshop on Statistical Machine Translation, pages 26–34,
Columbus, June 2008.

id	Translation	$\log(\mathbf{P_{TM}}(\mathbf{f}\|\mathbf{e}))$	$\log(\mathbf{P_{LM}}(\mathbf{e}))$	BLEU-2
$\mathbf{e}^1$	This is it	-1.2	-0.1	29.64
$\mathbf{e}^2$	This is small house	-0.2	-1.2	63.59
$\mathbf{e}^3$	This is miniscule building	-1.6	-0.9	31.79
$\mathbf{e}^4$	This is a small house	-0.1	-0.9	100.00
ref	This is a small house			

Table 1: Four hypothetical translations and their corresponding log model scores from a translation model $P_{TM}(f|e)$ and a language model $P_{LM}(e)$, along with their **BLEU-2** scores according to the given reference translation. The MERT error surface for these translations is given in figure 1.

tion of translations $\mathbf{E_w}$ onto some real valued loss, $\ell : \mathcal{E}^n \rightarrow \mathbb{R}$. For instance, in the experiments that follow, the loss corresponds to 1 minus the BLEU score assigned to $\mathbf{E_w}$ for a given collection of reference translations.

$$(\mathbf{E_w}, \mathbf{H_w}) = \underset{(\mathbf{E} \in \mathcal{E}^n, \mathbf{H} \in \mathcal{H}^n)}{\operatorname{argmax}} \mathbf{w} \cdot \Psi(\mathbf{E}, \mathbf{H}, \mathbf{F}) \quad (1)$$

Using n-best lists produced by a decoder to approximate $\mathcal{E}^n$ and $\mathcal{H}^n$, MERT searches for the weight vector $\mathbf{w}^*$ that minimizes the loss ℓ. Letting $\tilde{\mathbf{E}}_\mathbf{w}$ denote the result of the translation argmax w.r.t. the approximate hypothesis space, the MERT search is then expressed by equation (2). Notice the objective function being optimized is equivalent to the loss assigned by the automatic measure of translation quality, i.e. $\mathcal{O}(\mathbf{w}) = \ell(\tilde{\mathbf{E}}_\mathbf{w})$.

$$\mathbf{w}^* = \underset{\mathbf{w}}{\operatorname{argmin}}\, \ell(\tilde{\mathbf{E}}_\mathbf{w}) \quad (2)$$

After performing the parameter search, the decoder is then re-run using the weights $\mathbf{w}^*$ to produce a new set of n-best lists, which are then concatenated with the prior n-best lists in order to obtain a better approximation of $\mathcal{E}^n$ and $\mathcal{H}^n$. The parameter search given in (2) can then be performed over the improved approximation. This process repeats until either no novel entries are produced for the combined n-best lists or the weights change by less than some ϵ across iterations.

Unlike the objective functions associated with other popular learning algorithms, the objective $\mathcal{O}$ is piecewise constant over its entire domain. That is, while small perturbations in the weights, $\mathbf{w}$, will change the score assigned by $\mathbf{w} \cdot \Psi(\mathbf{e}, \mathbf{h}, \mathbf{f})$ to each triple, $(\mathbf{e}, \mathbf{h}, \mathbf{f})$, such perturbations will generally not change the ranking between the pair selected by the argmax, $(\mathbf{e}^*, \mathbf{h}^*) = \operatorname{argmax} \mathbf{w} \cdot \Psi(\mathbf{e}, \mathbf{h}, \mathbf{f})$, and any given competing pair $(\mathbf{e}', \mathbf{h}')$. However, at certain critical points, the score assigned to some competing pair $(\mathbf{e}', \mathbf{h}')$ will exceed that assigned to the prior winner $(\mathbf{e}^*_{w_{old}}, \mathbf{h}^*_{w_{old}})$. At this point, the pair returned by $\operatorname{argmax} \mathbf{w} \cdot \Psi(\mathbf{e}, \mathbf{h}, \mathbf{f})$ will change and loss ℓ will be evaluated using the newly selected $\mathbf{e}'$.

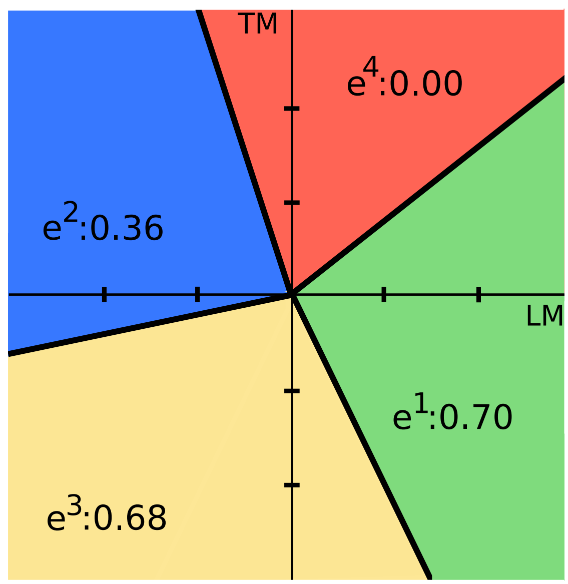

Figure 1: MERT objective for the translations given in table 1. Regions are labeled with the translation that dominates within it, i.e. $\operatorname{argmax} \mathbf{w} \cdot \Psi(\mathbf{e}, \mathbf{f})$, and with their corresponding objective values, $1 - \ell(\operatorname{argmax} \mathbf{w} \cdot \Psi(\mathbf{e}, \mathbf{f}))$.

This is illustrated in figure (1), which plots the MERT objective function for a simple model with two parameters, w_{tm} & w_{lm}, and for which the space of possible translations, $\mathcal{E}$, consists of the four sentences given in table 1[3]. Here, the loss ℓ is de-

[3]For this example, we ignore the latent variables, $\mathbf{h}$, associ-

fined as $1.0-\text{BLEU-2}(\mathbf{e})$. That is, ℓ is the difference between a perfect BLEU score and the BLEU score calculated for each translation using unigram and bi-gram counts.

The surface can be visualized as a collection of plateaus that all meet at the origin and then extend off into infinity. The latter property illustrates that the objective is scale invariant w.r.t. the weight vector $\mathbf{w}$. That is, since any vector $\mathbf{w}' = \lambda\mathbf{w}\ \forall_{\lambda>0}$ will still result in the same relative rankings of all possible translations according to $\mathbf{w} \cdot \Psi(\mathbf{e}, \mathbf{h}, \mathbf{f})$, such scaling will not change the translation selected by the $\arg\max$. At the boundaries between regions, the objective is undefined, as 2 or more candidates are assigned identical scores by the model. Thus, it is unclear what should be returned by the $\arg\max$ for subsequent scoring by ℓ.

Since the objective is piecewise constant, it cannot be minimized using gradient descent or even the sub-gradient method. Two applicable methods include downhill simplex and Powell's method (Press et al., 2007). The former attempts to find a local minimum in an n dimensional space by iteratively shrinking or growing an $n+1$ vertex simplex[4] based on the objective values of the current vertex points and select nearby points. In contrast, Powell's method operates by starting with a single point in weight space, and then performing a series of line minimizations until no more progress can be made. In this paper, we focus on line minimization based techniques, such as Powell's method.

2.1 Global minimum along a line

Even without gradient information, numerous methods can be used to find, or approximately find, local minima along a line. However, by exploiting the fact that the underlying scores assigned to competing hypotheses, $\mathbf{w} \cdot \Psi(\mathbf{e}, \mathbf{h}, \mathbf{f})$, vary linearly w.r.t. changes in the weight vector, $\mathbf{w}$, Och (2003) proposed a strategy for finding the global minimum along any given search direction.

The insight behind the algorithm is as follows. Let's assume we are examining two competing translation/derivation pairs, $(\mathbf{e}^1, \mathbf{h}^1)$ & $(\mathbf{e}^2, \mathbf{h}^2)$. Further, let's say the score assigned by the model to $(\mathbf{e}^1, \mathbf{h}^1)$ is greater than $(\mathbf{e}^2, \mathbf{h}^2)$, i.e. $\mathbf{w} \cdot \Psi(\mathbf{e}^1, \mathbf{h}^1, \mathbf{f}) > \mathbf{w} \cdot \Psi(\mathbf{e}^2, \mathbf{h}^2, \mathbf{f})$. Since the scores of the two vary linearly along any search direction, $\mathbf{d}$, we can find the point at which the model's relative preference for the competing pairs switches as $p = \frac{\mathbf{w}\cdot\Psi(\mathbf{e}^1,\mathbf{h}^1,\mathbf{f}) - \mathbf{w}\cdot\Psi(\mathbf{e}^2,\mathbf{h}^2,\mathbf{f})}{\mathbf{d}\cdot\Psi(\mathbf{e}^2,\mathbf{h}^2,\mathbf{f}) - \mathbf{d}\cdot\Psi(\mathbf{e}^1,\mathbf{h}^1,\mathbf{f})}$. At this particular point, we have the equality $(p\mathbf{d} + \mathbf{w}) \cdot \Psi(\mathbf{e}^1, \mathbf{h}^1, \mathbf{f}) = (p\mathbf{d} + \mathbf{w}) \cdot \Psi(\mathbf{e}^2, \mathbf{h}^2, \mathbf{f})$, or rather the point at which the scores assigned by the model to the candidates intersect along search direction $\mathbf{d}$[5]. Such points correspond to the boundaries between adjacent plateaus in the objective, as prior to the boundary the loss function ℓ is computed using the translation, $\mathbf{e}^1$, and after the boundary it is computed using $\mathbf{e}^2$.

To find the global minimum for a search direction $\mathbf{d}$, we move along $\mathbf{d}$ and for each plateau we

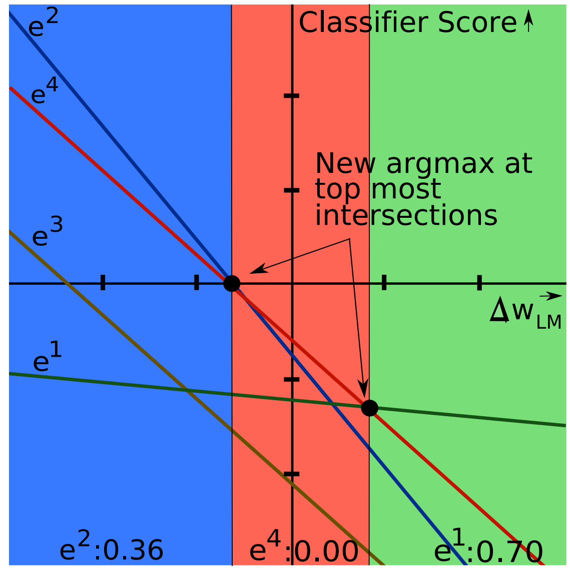

Figure 2: Illustration of how the model score assigned to each candidate translation varies during a line search along the coordinate direction w_{lm} with a starting point of $(w_{tm}, w_{lm}) = (1.0, 0.5)$. Each plotted line corresponds to the model score for one of the translation candidates. The vertical bands are labeled with the hypothesis that dominates in that region. The transitions between bands result from the dotted intersections between 1-best lines.

ated with the derivation of each $\mathbf{e}$ from the foreign sentence $\mathbf{f}$. If included, such variables would only change the graph in that multiple different derivations would be possible for each $\mathbf{e}^j$. If present, the graph could then include disjoint regions that all map to the same $\mathbf{e}^j$ and thus the same objective value.

[4] A simplex can be thought of as a generalization of a triangle to arbitrary dimensional spaces.

[5] Notice that, this point only exists if the slopes of the candidates' model scores along $\mathbf{d}$ are not equivalent, i.e. if $\mathbf{d} \cdot \Psi(\mathbf{e}^2, \mathbf{h}^2, \mathbf{f}) \neq \mathbf{d} \cdot \Psi(\mathbf{e}^1, \mathbf{h}^1, \mathbf{f})$.

Translation	m	b	1-best
e^1	-0.1	-1.25	(0.86,$+\infty$]
e^2	-1.2	-0.8	(-0.83,0.88)
e^3	-0.9	-2.05	n/a
e^4	-0.9	-0.55	[$-\infty$,-0.83]

Table 2: Slopes, m, intercepts, b, and 1-best ranges for the 4 translations given in table 1 during a line search along the coordinate w_{lm}, with a starting point of $(w_{tm}, w_{lm}) = (1.0, 0.5)$. This line search in illustrated in figure(2).

identify all the points at which the score assigned by the model to the current 1-best translation intersects the score assigned to competing translations. At the closest such intersection, we have a new 1-best translation. Moving to the plateau associated with this new 1-best, we then repeat the search for the nearest subsequent intersection. This continues until we know what the 1-best translations are for all points along $\mathbf{d}$. The global minimum can then be found by examining ℓ once for each of these.

Let's return briefly to our earlier example given in table 1. Starting at position $(w_{tm}, w_{lm}) = (1.0, 0.5)$ and searching along the w_{lm} coordinate, i.e. $(d_{tm}, d_{lm}) = (0.0, 1.0)$, table 2 gives the line search slopes, $m = \mathbf{d} \cdot \Psi(\mathbf{e}, \mathbf{h}, \mathbf{f})$, and intercepts, $b = \mathbf{w} \cdot \Psi(\mathbf{e}, \mathbf{h}, \mathbf{f})$, for each of the four candidate translations. Using the procedure just described, we can then find what range of values along $\mathbf{d}$ each candidate translation is assigned the highest relative model score. Figure 2 illustrates how the score assigned by the model to each of the translations changes as we move along $\mathbf{d}$. Each of the banded regions corresponds to a plateau in the objective, and each of the top most line intersections represents the transition from one plateau to the next. Note that, while the surface that is defined by the line segments with the highest classifier score for each region is convex, this is not a convex optimization problem as we are optimizing over the loss ℓ rather than classifier score.

Pseudocode for the line search is given in algorithm 1. Letting n denote the number of foreign sentences, $\mathbf{f}$, in a dataset, and having m denote the size of the individual n-best lists, $|l|$, the time complexity of the algorithm is given by $\mathcal{O}(nm^2)$. This is seen in that each time we check for the nearest intersection to the current 1-best for some n-best list l, we must calculate its intersection with all other candidate translations that have yet to be selected as the 1-best. And, for each of the n n-best lists, this may have to be done up to $m - 1$ times.

Algorithm 1 Och (2003)'s line search method to find the global minimum in the loss, ℓ, when starting at the point $\mathbf{w}$ and searching along the direction $\mathbf{d}$ using the candidate translations given in the collection of n-best lists $\mathcal{L}$.

Input: $\mathcal{L}$, $\mathbf{w}$, $\mathbf{d}$, ℓ
$\mathcal{I} \Leftarrow \{\}$
for $l \in \mathcal{L}$ **do**
  for $e \in l$ **do**
    m{e} $\Leftarrow$ e.features $\cdot$ $\mathbf{d}$
    b{e} $\Leftarrow$ e.features $\cdot$ $\mathbf{w}$
  end for
  best$_n \Leftarrow \text{argmax}_{e \in l}$ m{e} {b{e} breaks ties}
  loop
    best$_{n+1}$ = $\text{argmin}_{e \in l} \max\left(0, \frac{b\{best_n\} - b\{e\}}{m\{e\} - m\{best_n\}}\right)$
    intercept $\Leftarrow \max\left(0, \frac{b\{best_n\} - b\{best_{n+1}\}}{m\{best_{n+1}\} - m\{best_n\}}\right)$
    if intercept > 0 **then**
      add($\mathcal{I}$, intercept)
    else
      break
    end if
  end loop
end for
add($\mathcal{I}$, $\max(\mathcal{I}) + 2\epsilon$)
$i_{best} = \text{argmin}_{i \in \mathcal{I}} \text{eval}_\ell(\mathcal{L}, \mathbf{w} + (i - \epsilon) \cdot \mathbf{d})$
return $\mathbf{w} + (i_{best} - \epsilon) \cdot \mathbf{d}$

2.2 Search Strategies

In this section, we review two search strategies that, in conjunction with the line search just described, can be used to drive MERT. The first, Powell's method, was advocated by Och (2003) when MERT was first introduced for statistical machine translation. The second, which we call Koehn-coordinate descent (KCD)[6], is used by the MERT utility packaged with the popular Moses statistical machine translation system (Koehn et al., 2007).

[6]Moses uses David Chiang's CMERT package. Within the source file mert.c, the function that implements the overall search strategy, optimize_koehn(), is based on Philipp Koehn's Perl script for MERT optimization that was distributed with Pharaoh.

2.2.1 Powell's Method

Powell's method (Press et al., 2007) attempts to efficiently search the objective by constructing a set of mutually non-interfering search directions. The basic procedure is as follows: (i) A collection of search directions is initialized to be the coordinates of the space being searched; (ii) The objective is minimized by looping through the search directions and performing a line minimization for each; (iii) A new search direction is constructed that summarizes the cumulative direction of the progress made during step (ii) (i.e., $\mathbf{d}_{new} = \mathbf{w}_{pre_{ii}} - \mathbf{w}_{post_{ii}}$). After a line minimization is performed along $\mathbf{d}_{new}$, it is used to replace one of the existing search directions. (iv) The process repeats until no more progress can be made. For a quadratic function of n variables, this procedure comes with the guarantee that it will reach the minimum within n iterations of the outer loop. However, since Powell's method is usually applied to non-quadratic optimization problems, a typical implementation will forego the quadratic convergence guarantees in favor of a heuristic scheme that allows for better navigation of complex surfaces.

2.2.2 Koehn's Coordinate Descent

KCD is a variant of coordinate descent that, at each iteration, moves along the coordinate which allows for the most progress in the objective. In order to determine which coordinate this is, the routine performs a trial line minimization along each. It then updates the weight vector with the one that it found to be most successful. While much less sophisticated that Powell, our results indicate that this method may be marginally more effective at optimizing the MERT objective[7].

3 Extensions

In this section we present and motivate two novel extensions to MERT. The first is a stochastic alternative to the Powell and KCD search strategies, while the second is an efficient method for regularizing the objective.

[7]While we are not aware of any previously published results that demonstrate this, it is likely that we were not the first to make this discovery as even though Moses' MERT implementation includes a vestigial implementation of Powell's method, the code is hardwired to call optimize_koehn rather than the routine for Powell.

3.1 Random Search Directions

One significant advantage of Powell's algorithm over coordinate descent is that it can optimize along diagonal search directions in weight space. That is, given a model with a dozen or so features, it can explore gains that are to be had by simultaneously varying two or more of the feature weights. In general, the diagonals that Powell's method constructs allow it to walk objective functions more efficiently than coordinate descent (Press et al., 2007). However, given that we have a line search algorithm that will find the global minima along any given search direction, diagonal search may be of even more value. That is, similar to ridge phenomenon that arise in traditional hill climbing search, it is possible that there are points in the objective that are the global minimum along any given coordinate direction, but are not the global minimum along diagonal directions.

However, one substantial disadvantage for Powell is that the assumptions it uses to build up the diagonal search directions do not hold in the present context. Specifically, the search directions are built up under the assumption that near a minimum the surface looks approximately quadratic and that we are performing local line minimizations within such regions. However, since we are performing global line minimizations, it is possible for the algorithm to jump from the region around one minima to another. If Powell's method has already started to tune its search directions for the prior minima, it will likely be less effective in its efforts to search the new region. To this extent, coordinate descent will be more robust than Powell as it has no assumptions that are violated when such a jump occurs.

One way of salvaging Powell's algorithm in this context would be to incorporate additional heuristics that detect when the algorithm has jumped from the region around one minima to another. When this occurs, the search directions could be reset to the coordinates of the space. However, we opt for a simpler solution, which like Powell's algorithm performs searches along diagonals in the space, but that like coordinate descent is sufficiently simple that the algorithm will not be confused by sudden jumps between regions.

Specifically, the search procedure chooses directions at random such that each component

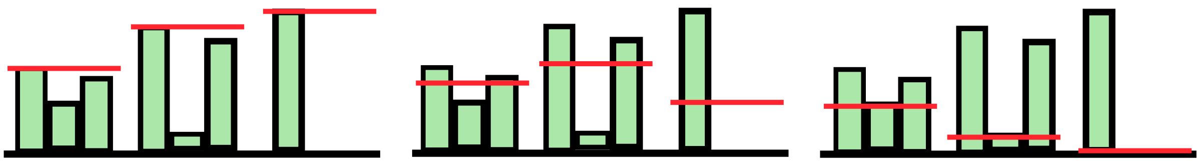

Figure 3: Regularization during line search - using, from left to right: (i) the maximum loss of adjacent plateaus, (ii) the average loss of adjacent plateaus, (iii) no regularization. Each set of bars represents adjacent plateaus along the line being searched, with the height of the bars representing their associated loss. The vertical lines indicate the surrogate loss values used for the center region under each of the schemes (i-iii).

is distributed according to a Gaussian[8], $\mathbf{d}$ s.t. $d_i \sim N(0,1)$. This allows the procedure to minimize along diagonal search directions, while making essentially no assumptions regarding the characteristics of the objective or the relationship between a series of sequential line minimizations. In the results that follow, we show that, perhaps surprisingly, this simple procedure outperforms both KCD and Powell's method.

3.2 Regularization

One potential drawback of MERT, as it is typically implemented, is that it attempts to find the best possible set of parameters for a training set without making any explicit efforts to find a set of parameters that can be expected to generalize well. For example, let's say that for some objective there is a very deep but narrow minima that is surrounded on all sides by very bad objective values. That is, the BLEU score at the minima might be 39.1 while all surrounding plateaus have a BLEU score that is < 10. Intuitively, such a minima would be a very bad solution, as the resulting parameters would likely exhibit very poor generalization to other data sets. This could be avoided by regularizing the surface in order to eliminate such spurious minima.

One candidate for performing such regularization is the continuous approximation of the MERT objective, $\mathcal{O} = \mathbb{E}_{p_{\mathbf{w}}}(\ell)$. Och (2003) claimed that this approximation achieved essentially equivalent performance to that obtained when directly using the loss as the objective, $\mathcal{O} = \ell$. However, Zens et al. (2007) found that $\mathcal{O} = \mathbb{E}_{p_{\mathbf{w}}}(\ell)$ achieved substantially better test set performance than $\mathcal{O} = \ell$, even though it performs slightly worse on the data used to train the parameters. Similarly, Smith and Eisner (2006) reported test set gains for the related technique of minimum risk annealing, which incorporates a temperature parameter that trades off between the smoothness of the objective and the degree it reflects the underlying piecewise constant error surface. However, the most straightforward implementation of such methods requires a loss that can be applied at the sentence level. If the evaluation metric of interest does not have this property (e.g. BLEU), the loss must be approximated using some surrogate, with successful learning then being tied to how well the surrogate captures the critical properties of the underlying loss.

The techniques of Zens et al. (2007) & Smith and Eisner (2006) regularize by implicitly smoothing over nearby plateaus in the error surface. We propose an alternative scheme that operates directly on the piecewise constant objective and that mitigates the problem of spurious local minima by explicitly smoothing over adjacent plateaus during the line search. That is, when assessing the desirability of any given plateau, we examine a fixed window w of adjacent plateaus along the direction being searched and combine their evaluation scores. We explore two combination methods, max and $average$. The former, max, assigns each plateau an objective value that is equal to the maximum objective value in its surrounding window, while $average$ assigns a plateau an objective value that is equal to its window's average. Figure 3 illustrates both methods for regularizing the plateaus and contrasts them with the case where no regularization is used. Notice that, while both methods discount spurious pits in the objective, $average$ still does place some value on isolated deep plateaus, and max discounts them completely.

Note that one potential weakness of this scheme is the value assigned by the regularized objective to any given point differs depending on the direction being searched. As such, it has the potential to wreak havoc on methods such as Powell's, which effectively attempt to learn about the curvature of the

[8] However, we speculate that similar results could be obtained using a uniform distribution over $(-1, 1)$

objective from a sequence of line minimizations.

4 Experiments

Three sets of experiments were performed. For the first set, we compare the performance of Powell's method, KCD, and our novel stochastic search strategy. We then evaluate the performance of all three methods when the objective is regularized using the average of adjacent plateaus for window sizes varying from 3 to 7. Finally, we repeat the regularization experiment, but using the maximum objective value from the adjacent plateaus. These experiments were performed using the Chinese English evaluation data provided for NIST MT eval 2002, 2003, and 2005. MT02 was used as a dev set for MERT learning, while MT03 and MT05 were used as our test sets.

For all experiments, MERT training was performed using n-best lists from the decoder of size 100. During each iteration, the MERT search was performed once with a starting point of the weights used to generate the most recent set of n-best lists and then 5 more times using randomly selected starting points[9]. Of these, we retain the weights from the search that obtained the lowest objective value. Training continued until either decoding produced no novel entries for the combined n-best lists or none of the parameter values changed by more than 1e-5 across subsequent iterations.

4.1 System

Experiments were run using a right-to-left beam search decoder that achieves a matching BLEU score to Moses (Koehn et al., 2007) over a variety of data sets. Moreover, when using the same underlying model, the two decoders only produce translations that differ by one or more words 0.2% of the time. We made use of a stack size of 50 as it allowed for faster experiments while only performing modestly worse than a stack of 200. The distortion limit was set to 6. And, we retrieved 20 translation options for each unique source phrase.

Our phrase table was built using $1,140,693$ sentence pairs sampled from the GALE Y2 training data. The Chinese data was word segmented using the GALE Y2 retest release of the Stanford CRF segmenter (Tseng et al., 2005). Phrases were extracted using the typical approach described in Koehn et al. (2003) of running GIZA++ (Och & Ney, 2003) in both directions and then merging the alignments using the grow-diag-final heuristic. From the merged alignments we also extracted a bidirectional lexical reordering model conditioned on the source and the target phrases (Tillmann, 2004) (Koehn et al., 2007). A 5-gram language model was created using the SRI language modeling toolkit (Stolcke, 2002) and trained using the Gigaword corpus and English sentences from the parallel data.

[9]Only 5 random restarts were used due to time constraints. Ideally, a sizable number of random restarts should be used in order to minimize the degree to which the results are influenced by some runs receiving starting points that are better in general or perhaps better/worse w.r.t. their specific optimization strategy.

Method	**Dev MT02**	**Test MT03**	**Test MT05**
KCD	30.967	30.778	29.580
Powell	30.638	30.692	29.780
Random	31.681	31.754	30.191

Table 3: BLEU scores obtained by models trained using the three different parameter search strategies: Powell's method, KCD, and stochastic search.

5 Results

As illustrated in table 3, Powell's method and KCD achieve a very similar level of performance, with KCD modestly outperforming Powell on the MT03 test set while Powell modestly outperforms coordinate descent on the MT05 test set. Moreover, the fact that Powell's algorithm did not perform better than KCD on the training data[10], and in fact actually performed modestly worse, suggests that Powell's additional search machinery does not provide much benefit for MERT objectives.

Similarly, the fact that the stochastic search obtains a much higher dev set score than either Powell or KCD indicates that it is doing a better job of optimizing the objective than either of the two alternatives. These gains suggest that stochastic search does make better use of the global minimum line search than the alternative methods. Or, alternatively, it strengthens the claim that the method succeeds at combining one of the critical strengths

[10]This indicates that Powell failed to find a deeper minima in the objective, since recall that the unregularized objective is equivalent to the model's dev set performance.

Method	**Window Avg**	**Dev MT02**	**Test MT03**	**Test MT05**	**Method**	**Window Max**	**Dev MT02**	**Test MT03**	**Test MT05**
Coordinate	none	30.967	30.778	29.580	Coordinate	none	30.967	30.778	29.580
	3	31.665	**31.675**	**30.266**		3	31.536	**31.927**	**30.334**
	5	31.317	31.229	**30.182**		5	31.484	**31.702**	29.687
	7	31.205	**31.824**	30.149		7	31.627	31.294	**30.199**
Powell	none	30.638	30.692	29.780	Powell	none	30.638	30.692	29.780
	3	31.333	31.412	29.890		3	31.428	30.944	29.598
	5	31.748	**31.777**	**30.334**		5	31.407	**31.596**	30.090
	7	31.249	**31.571**	30.161		7	30.870	30.911	29.620
Random	none	31.681	31.754	30.191	Random	none	31.681	31.754	30.191
	3	31.548	31.778	30.263		3	31.179	30.898	29.529
	5	31.336	31.647	30.415		5	30.903	31.666	29.963
	7	30.501	29.336	28.372		7	31.920	31.906	30.674

Table 4: BLEU scores obtained when regularizing using the average loss of adjacent plateaus, left, and the maximum loss of adjacent plateaus, right. The none entry for each search strategy represents the baseline where no regularization is used. Statistically significant test set gains, $p < 0.01$, over the respective baselines are in bold face.

of Powell's method, diagonal search, with coordinate descent's robustness to the sudden jumps between regions that result from global line minimization. Using an approximate randomization test for statistical significance (Riezler & Maxwell, 2005), and with KCD as a baseline, the gains obtained by stochastic search on MT03 are statistically significant ($p = 0.002$), as are the gains on MT05 ($p = 0.005$).

Table 4 indicates that performing regularization by either averaging or taking the maximum of adjacent plateaus during the line search leads to gains for both Powell's method and KCD. However, no reliable additional gains appear to be had when stochastic search is combined with regularization.

It may seem surprising that the regularization gains for Powell & KCD are seen not only in the test sets but on the dev set as well. That is, in typical applications, regularization slightly decreases performance on the data used to train the model. However, this trend can in part be accounted for by the fact that during training, MERT is using n-best lists for objective evaluations rather than the more expensive process of running the decoder for each point that needs to be checked. As such, during each iteration of training, the decoding performance of the model actually represents its generalization performance relative to what was learned from the n-best lists created during prior iterations. Moreover, better generalization from the prior n-best lists can also help drive subsequent learning as there will then be more high quality translations on the n-best lists used for future iterations of learning. Additionally, regularization can reduce search errors by reducing the risk of getting stuck in spurious low loss pits that are in otherwise bad regions of the space.

6 Conclusions

We have presented two methods for improving the performance of MERT. The first is a novel stochastic search strategy that appears to make better use of Och (2003)'s algorithm for finding the global minimum along any given search direction than either coordinate descent or Powell's method. The second is a simple regularization scheme that leads to performance gains for both coordinate descent and Powell's method. However, no further gains are obtained by combining the stochastic search with regularization of the objective.

One quirk of the regularization scheme presented here is that the regularization applied to any given point in the objective varies depending upon what direction the point is approached from. We are currently looking at other similar regularization schemes that maintain consistent objective values regardless of the search direction.

Acknowledgments

We extend our thanks to our three anonymous reviewers, particularly for the depth of analysis provided. This paper is based on work funded in part by the Defense Advanced Research Projects Agency through IBM.

References

Koehn, P., Hoang, H., Birch, A., Callison-Burch, C., Federico, M., Bertoldi, N., Cowan, B., Shen, W., Moran, C., Zens, R., Dyer, C., Bojar, O., Constantin, A., & Herbst, E. (2007). Moses: Open source toolkit for statistical machine translation. *In ACL*.

Koehn, P., Och, F. J., & Marcu, D. (2003). Statistical phrase-based translation. *In HLT-NAACL*.

Och, F.-J. (2003). Minimum error rate training in statistical machine translation. *In ACL*.

Och, F. J., & Ney, H. (2002). Discriminative training and maximum entropy models for statistical machine translation. *In ACL*.

Och, F. J., & Ney, H. (2003). A systematic comparison of various statistical alignment models. *Computational Linguistics*, *29*, 19–51.

Papineni, K., Roukos, S., Ward, T., & Zhu, W.-J. (2002). Bleu: a method for automatic evaluation of machine translation. *In ACL*.

Press, W. H., Teukolsky, S. A., Vetterling, W. T., & Flannery, B. P. (2007). *Numerical recipes 3rd edition: The art of scientific computing*. Cambridge University Press.

Riezler, S., & Maxwell, J. T. (2005). On some pitfalls in automatic evaluation and significance testing for mt. *In ACL*.

Smith, D. A., & Eisner, J. (2006). Minimum risk annealing for training log-linear models. *In ACL*.

Stolcke, A. (2002). Srilm – an extensible language modeling toolkit. *In ICSLP*.

Tillmann, C. (2004). A unigram orientation model for statistical machine translation. *In ACL*.

Tseng, H., Chang, P., Andrew, G., Jurafsky, D., & Manning, C. (2005). A conditional random field word segmenter for sighan bakeoff 2005. *In SIGHAN Workshop on Chinese Language Processing*.

Zens, R., Hasan, S., & Ney, H. (2007). A systematic comparison of training criteria for statistical machine translation. *In EMNLP*.