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 strategies1. 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 F be a collection of foreign sentences to be translated, with individual sentences f0, f1, ..., fn. For each fi, the surface form of an individual candidate translation is given by ei with hidden state hi associated with the derivation of ei from fi. Each ei is drawn from E, which represents all possible strings our translation system can produce. The (ei, hi, fi) triples are converted into vectors of m feature functions by Ψ : E × H × F → Rm whose dot product with the weight vector w assigns a score to each triple. The idealized translation process then is to find the highest scoring pair (ei, hi) for each fi, or rather (ei, hi) = argmax (e ∈ E, h ∈ H) w · Ψ(e, h, f).

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

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

2Here, the translation of the entire data set is treated as a single structured prediction problem using the feature function vector Ψ(E, H, F) = \sum_i^n Ψ(e_i, h_i, f_i)
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.

| id | Translation                  | log($P_{TM}(f|e)$) | log($P_{LM}(e)$) | BLEU-2 |
|----|-----------------------------|---------------------|------------------|--------|
| e₁ | This is it                  | -1.2               | -0.1             | 29.64  |
| e₂ | This is small house         | -0.2               | -1.2             | 63.59  |
| e₃ | This is miniscule building  | -1.6               | -0.9             | 31.79  |
| e₄ | This is a small house       | -0.1               | -0.9             | 100.00 |

For this example, we ignore the latent variables, $h$, associating $\tilde{e}$ with the reference translations. The score assigned to $E_w$ for a given collection of reference translations is piecewise constant over its entire domain. That is, while small perturbations in the weights, $w$, will change the ranking between the pair selected by the argmax, $\langle e^*, h^* \rangle = \argmax_w \Psi(e, h, f)$, and any given competing pair $\langle e', h' \rangle$. However, at certain critical points, the score assigned to some competing pair $\langle e', h' \rangle$ will exceed that assigned to the prior winner $\langle e_{old}, h_{old} \rangle$. At this point, the pair returned by $\argmax_w \Psi(e, h, f)$ will change and loss $\ell$ will be evaluated using the newly selected $e'$.
fined as $1.0-\text{BLEU-2}(e)$. That is, $\ell$ 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 $w$. That is, since any vector $w'=\lambda w \forall \lambda>0$ will still result in the same relative rankings of all possible translations according to $w \cdot \Psi(e, h, f)$, such scaling will not change the translation selected by the argmax. 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 argmax for subsequent scoring by $\ell$.

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\footnote{A simplex can be thought of as a generalization of a triangle to arbitrary dimensional spaces.} 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, $w \cdot \Psi(e, h, f)$, vary linearly w.r.t. changes in the weight vector, $w$, 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, $(e^1, h^1)$ & $(e^2, h^2)$. Further, let’s say the score assigned by the model to $(e^1, h^1)$ is greater than $(e^2, h^2)$, i.e. $w \cdot \Psi(e^1, h^1, f) > w \cdot \Psi(e^2, h^2, f)$. Since the scores of the two vary linearly along any search direction, $d$, we can find the point at which the model’s relative preference for the competing pairs switches as $p = \frac{w \cdot \Psi(e^1, h^1, f) - w \cdot \Psi(e^2, h^2, f)}{d \cdot \Psi(e^2, h^2, f) - d \cdot \Psi(e^1, h^1, f)}$. At this particular point, we have the equality $(pd + w) \cdot \Psi(e^1, h^1, f) = (pd + w) \cdot \Psi(e^2, h^2, f)$, or rather the point at which the scores assigned by the model to the candidates intersect along search direction $d$. Such points correspond to the boundaries between adjacent plateaus in the objective, as prior to the boundary the loss function $\ell$ is computed using the translation, $e^1$, and after the boundary it is computed using $e^2$.

To find the global minimum for a search direction $d$, we move along $d$ 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 $(w_{lm}, w_{lm}) = (1.0, 0.5)$. 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.](image-url)
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 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 \((w_{lm}, w_{lm}) = (1.0, 0.5)\) and searching along the \(w_{lm}\) coordinate, i.e. \((d_{lm}, d_{lm}) = (0.0, 1.0)\), Table 2 gives the line search slopes, \(m = d \cdot \Psi(e, h, f)\), and intercepts, \(b = w \cdot \Psi(e, h, f)\), for each of the four candidate translations. Using the procedure just described, we can then find what range of values along 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 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, \(f\), 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 \(O(nm^2)\). 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.

### 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), is used by the MERT utility packaged with the popular Moses statistical machine translation system (Koehn et al., 2007).

#### Algorithm 1

<table>
<thead>
<tr>
<th>Translation</th>
<th>(m)</th>
<th>(b)</th>
<th>1-best</th>
</tr>
</thead>
<tbody>
<tr>
<td>(e^1)</td>
<td>-0.1</td>
<td>-1.25</td>
<td>(0.86, +(∞))</td>
</tr>
<tr>
<td>(e^2)</td>
<td>-1.2</td>
<td>-0.8</td>
<td>(-0.83, 0.88)</td>
</tr>
<tr>
<td>(e^3)</td>
<td>-0.9</td>
<td>-2.05</td>
<td>n/a</td>
</tr>
<tr>
<td>(e^4)</td>
<td>-0.9</td>
<td>-0.55</td>
<td>[-(∞), -0.83]</td>
</tr>
</tbody>
</table>

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 \((w_{lm}, w_{lm}) = (1.0, 0.5)\). This line search in illustrated in Figure 2.

\[ \text{Algorithm 1: Och (2003)’s line search method to find the global minimum in the loss, } \ell, \text{ when starting at the point } w \text{ and searching along the direction } d \text{ using the candidate translations given in the collection of } n\text{-best lists } L. \]

\[ \text{Input: } L, w, d, \ell \]

\[ I \leftarrow \{\} \]

for \(l \in L\) do

for \(e \in l\) do

\[ m\{e\} \leftarrow e\text{.features} \cdot d \]

\[ b\{e\} \leftarrow e\text{.features} \cdot w \]

end for

\[ \text{best}_n \leftarrow \text{argmax}_{e \in l} \{ m\{e\} \{ b\{e\} \text{ breaks ties} \} \} \]

loop

\[ \text{best}_{n+1} = \text{argmin}_{e \in l} \max \left( 0, \frac{b\{\text{best}_n\} - b\{e\}}{m\{e\} - m\{\text{best}_n\}} \right) \]

\[ \text{intercept} \leftarrow \max \left( 0, \frac{b\{\text{best}_n\} - b\{\text{best}_{n+1}\}}{m\{\text{best}_{n+1}\} - m\{\text{best}_n\}} \right) \]

if intercept > 0 then

add(\(I\), intercept)

else

break

end if

end loop

end for

\[ \text{add}(I, \text{max}(I) + 2\epsilon) \]

\[ i_{\text{best}} = \text{argmin}_{i \in I} \text{eval}_\ell(L, w + (i - \epsilon) \cdot d) \]

return \(w + (i_{\text{best}} - \epsilon) \cdot d\)
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., $d_{new} = w_{pre} - w_{post}$). After a line minimization is performed along $d_{new}$, 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.

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...
is distributed according to a Gaussian, \( d \sim N(0, 1) \). 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, \( O = E_{pw}(\ell) \). Och (2003) claimed that this approximation achieved essentially equivalent performance to that obtained when directly using the loss as the objective, \( O = \ell \). However, Zens et al. (2007) found that \( O = E_{pw}(\ell) \) achieved substantially better test set performance than \( O = \ell \), 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, \( \text{max} \) and \( \text{average} \). The former, \( \text{max} \), assigns each plateau an objective value that is equal to the maximum objective value in its surrounding window, while \( \text{average} \) 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, \( \text{average} \) still does place some value on isolated deep plateaus, and \( \text{max} \) 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
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. 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 bi-directional 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.

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, 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

<table>
<thead>
<tr>
<th>Method</th>
<th>Dev MT02</th>
<th>Test MT03</th>
<th>Test MT05</th>
</tr>
</thead>
<tbody>
<tr>
<td>KCD</td>
<td>30.967</td>
<td>30.778</td>
<td>29.580</td>
</tr>
<tr>
<td>Powell</td>
<td>30.638</td>
<td>30.692</td>
<td>29.780</td>
</tr>
<tr>
<td>Random</td>
<td>31.681</td>
<td>31.754</td>
<td>30.191</td>
</tr>
</tbody>
</table>

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

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.

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


34