Considerations in Maximum Mutual Information and Minimum Classification Error training for Statistical Machine Translation

Ashish Venugopal* and **Stephan Vogel****
Language Technologies Institute
Carnegie Mellon University
{ashishv;stephan.vogel}@cs.cmu.edu

Abstract. Discriminative training methods are used in statistical machine translation to effectively introduce and combine additional knowledge sources within the translation process. Although these methods are described in the accompanying literature and comparative studies are available for speech recognition, additional considerations are introduced when applying discriminative training to statistical machine translation. In this paper we pay special attention to the comparison and formalization of discriminative training criteria and their respective optimization methods with the goal of improving translation performance measured by the corpus level BLEU metric for a Viterbi beam based decoder. We frame this work within the current trends in discriminative training and present reproducible results that highlight the potential as well as shortcomings of N-Best list based discriminative training.

1. Introduction

Statistical machine translation, like other natural language process tasks, has developed a set of unique evaluation metrics that go beyond simply evaluating the number of sentence errors that a system makes on a test set. While debates continue regarding the relative value of each competing metric, the BLEU [Papeneni, 2001] and NIST [Doddington, 2002] scores (which consider system performance at the corpus level) have shown their effectiveness in driving the development of statistical machine translation systems. These metrics have highlighted the need for more expressive models of translation and a framework to introduce additional knowledge sources within the translation process. The direction translation [Och, 2002] approach [Brown et al, 1993] delivers this framework, and provides a necessary formalism to the process of combining and optimizing additional knowledge sources. Discriminative training considers competing candidate translations from an N-Best list is used to find appropriate scaling factors for each additional knowledge source. The goal is to find scaling factors that improve the metric performance of the candidate translation chosen by the decoder [Vogel, 2003]. In this paper, we consider these scaling factors within the decoding process rather than as a post processing reranking step, thereby creating additional considerations regarding the stability of the scaling factors. The choice of evaluation metric, the nature of the additional knowledge sources within the decoding process and the implementation decisions taken in each component, determine the effectiveness of each discriminative method.

This paper will focus on comparing the formalism and practical considerations involved with deploying Maximum Mutual Information [Bassat,1982] and Minimum Classification Error [Huang, Katagiri, 1992] training within a statistical machine translation context. We begin by framing the discriminative training task for statistical machine translation and survey directions of active research in the field. We discuss the impact that the corpus level BLEU score has on the discriminative training criteria and the implementation requirements for optimization methods that accommodate for such metrics. We describe the process of generating N-Best lists from a Viterbi decoding using partial and full translation based knowledge sources and then merging these lists across iterations along with experimental results on widely distributed training and test data sets. We conclude with a discussion of future work and potentially promising directions in discriminative training.

2. Direct Statistical Machine Translation

Statistical machine translation presents the task of finding a target language ("English") sequence of word tokens e = $e_1 \ldots e_S$ that is the translation for a source language ("French") sequence f = $f_1 \ldots f_T$. A zero-one loss function would suggest that a decision rule that selects the English sentence that has the highest conditional probability, choosing from the set of all possible target language sequences E as shown below.

$$e* = \arg\max_e P(e \mid f) \qquad (1)$$

where $P(e \mid f)$ refers to the true conditional distribution of e given f. Using this decision rule to select from within the search space of all possible candidate translations minimizes the number of decision errors made under a zero-one loss function (which implies there is one correct translation, and several incorrect translations). The decoder [Vogel, 2003] performs this search using a parameterized estimate $P_\theta(e \mid f)$ of $P(e \mid f)$. The search is kept tractable by aggressive pruning based on the estimated model. It is clear that this decision rule does not explicitly model performance on an evaluation metric, but rather leverages the effectiveness of estimate $P_\theta(e \mid f)$ to rank competing candidate sequences. [Kumar, Byrnes, 2004] propose a Minimum Bayes Risk (MBR) decoding process that explicitly minimizes the expected value of the loss according to an evaluation metric for a training set. As stated in [Kumar,Byrnes, 2004], performing the search process and computing the expectation of the loss over the true distributions is computationally prohibitive and they limit the use of their MBR decoder to re-ranking an N-Best list. [Shen, 2004] also proposes discriminative re-ranking on N-Best lists that focus on separating "good" and "bad" translations according to an evaluation metric. Our discussion will not focus on N-Best list re-ranking, but instead, will investigate methods that use the N-Best list as an approximation of the search environment within the decoder. We limit our scope to MAP decoders and determining model scaling factors $\theta = \theta_1 \ldots \theta_M$ for $P_\theta(e \mid f)$ that improve the decoder's performance on the BLEU metric.

Under the source channel approach presented in [Brown et al, 1993], the decision rule (1) decomposed using the Bayes rules into $P(f \mid e)$ and $P(e)$, which can be individually estimated. These models are usually combined as a log linear model with scaling factors that are tuned to bias the performance of the system towards a particular evaluation metric. [Och, 2002] proposed modeling the direct translation probability $P_\theta(e \mid f)$ directly, allowing for extensions to the two model approach without loss of generality using the exponential model

$$P_\theta(e \mid f) = \frac{e^{\sum_{m=1}^{M} \theta_m * h_m(e,f)}}{\sum_{k=1}^{|E|} e^{\sum_{m=1}^{M} \theta_m * h_m(e,f)}}$$

where h is a model feature score that represents some relationship between e, f. In the source channel model we could use log forms of the translation and language models as model features. Henceforth we will omit the individual model subscripts and simply refer to $\theta.h(e, f)$ to represent the linear combination of all scaling factors and their respective models features. We now discuss discriminative methods to find θ on a training corpus such that decoding using the decision rule in equation (1) to decode a test set will improve performance as measured by the BLEU metric.

3. Discriminative Training

[Normandin 1994] provided empirical evidence that discriminative training criteria could better recover from situations where incorrect model assumptions are made, since these criteria attempt to separate the class conditional probabilities of the correct class $e*$ from the alternative classes $e' \in E$ from the N-Best list.

3.1. Maximum Mutual Information

The Maximum Mutual Information [Bassat, 1982] uses the evaluation metric to label "correct" classes $e^+ \in E$ and attempts to find θ for $P_\theta(e \mid f)$ such that these correct classes are separated from the incorrect classes $e^- \in E$. MMI defines the objective function over a set of N training source-target language sequence pairs.

$$F_{MMI} = \frac{1}{N}\sum\nolimits_{n=1}^{N} \log P_\theta\left(e_n^+ \mid f_n\right)$$

Under the direct estimation approach, it is unnecessary to decompose $P_\theta(e \mid f)$ further, and this method reduces to the conditional maximum likelihood criteria allowing simple gradient based optimization techniques. The discrimination is implicit in $P_\theta(e \mid f)$ since in log form we are separating the scores of the metric specified "correct" translations and the competing candidates from the N-Best list.

$$\begin{aligned} F_{MMI} &= \frac{1}{N}\sum\nolimits_{n=1}^{N}\left[\theta.h(e_n^+, f_n) - \log \sum_{e_n^k \in E_n} e^{\theta.h(e_n^k, f_n)}\right] \\ \theta^* &= \arg\max\nolimits_\theta F_{MMI} \end{aligned}$$

F_{MMI} is a smooth, differentiable function with gradient

$$\frac{\delta F_{MMI}}{\delta m} = \frac{1}{N}\sum\nolimits_{n=1}^{N}\left[h_m(e_n^+, f_n) - \sum_{e_n^k \in E_n} \frac{e^{\theta.h(e_n^k, f_n)}}{\sum_{e_n^k \in E_n} e^{\theta.h(e_n^k, f_n)}} .h_m(e_n^k, f_n)\right]$$

with respect to each dimension m in θ. Several techniques for this kind of optimization are discussed in [Press et al, 2002].

3.2. Minimum Classification Error

The Minimum Classification Error [Juang, Katagiri, 1992] criteria attempts to minimize the empirical error (as determined by the evaluation metric) of the decision rule. To explicitly model this condition, we can define our MCE criterion as shown in [Och,2003] as…

$$\begin{aligned} F_{MCE} &= \frac{1}{N}\sum\nolimits_{n=1}^{N}\left[Error(e_n^*, r_n)\right] \\ e_n^* &= \arg\max\nolimits_{e_n^k \in E_n} \theta.h(e_n^k, f_n) \\ \theta^* &= \arg\min\nolimits_\theta F_{MCE} \end{aligned} \qquad (2)$$

where $Error(e_n^*, r_n)$ is a function that assigns an error to the selected candidate sequence e_n^* with respect to a reference translation for r_n that is available for each f_n. Note that the decision rules used by both methods is the same, their difference lies in the objective function used to train the scaling factors.

The MCE criterion can be smoothed into continuous, differentiable functions that can be optimized with respect to θ using conjugate gradient descents. This is usually accomplished by using a "softmax" operation to replace argmax as shown in [Schlueter, 2001] and a smooth error function, usually a sigmoid function to replace a zero-one loss function. The form shown in equation (2) however, would typically require gradient free optimization techniques such as Powell's method or the Snelder-Mead simplex method as described in [Press]. [Och2003] shows that a much simpler optimization method is available that leverages the form of $P_\theta(e \mid f)$ to optimize each dimension of θ much more efficiently. We provide the details later in this paper. The criterion in equation (2) can be compared to Falsifying Training as described in [Shleuter] where $\alpha = \infty$ and the $Error(e_n^*, r_n)$ is the smoothing function (albeit not smooth). This form allows us to accurately estimate the empirical error of the same decision rule (2) used in the decoder on the training data and optimize θ to minimize this error. Although this method is not guaranteed to converge on a globally optimal θ^*, [Schlueter, 2001] shows that the MCE criterion achieves a tighter error bound on the true Bayes error rate.

[Zens,Ney, 2003] propose an alternative method where the error surface is evaluated using only e_n^* without regard for the alternatives available in the N-Best list. The empirical error on the training data is estimated using the top candidate only. [Zens,Ney, 2003] present an iterative algorithm via the Simplex method described in [Press et al, 2002], generating a locally optimal final parameter set.

While the N-Best list based methods explore a larger region of the error surface after decoding the training set, they select e_n^* from the N-Best list only, which is an approximation to the true space of all candidate target language sequences. This approximation is potentially a function of several pruning and recombination parameters that drive the decoder through the search space and influence the final N-Best list. Since the method described in [Zens,Ney] works only with the top candidates, the decision rule used to search the space is a more accurate representation of the decision process used in the decoding and optimization can operate on all parameters that play a role in the decoding, rather than just those that play a role in $P_\theta(e \mid f)$. The disadvantage of this method lies in the computational cost of repeatedly decoding and evaluating a training set especially when the Simplex method is typically one of the slowest to converge [Press et al, 2002].

We will now consider the MMI and MCE criteria (both N-Best list based methods) and specific considerations when applying them to statistical machine translation.

4. Effect of the Evaluation Metric

Both the MMI and MCE methods are inherently related to the choice of metric used to evaluation candidate translation. MMI uses the metric to label "correct" and "incorrect" translations, while the MCE explicitly evaluates candidates in the N-Best list that are chosen by the decision rule to minimize the error on these choices. Several evaluation metrics including word error rate, multi reference word error rate, BLEU and NIST are commonly used in statistical machine translation and [Och, 2003] presents empirical evidence that optimizing $P_\theta(e \mid f)$ using a particular metric will yield the most improved results when evaluating the decision rule using the same metric. BLEU and NIST however, are evaluated at the corpus level and are not additive over individual sentences. This makes choosing the "correct" translation from the N-Best list for MMI difficult and $Error(e_n^*, r_n)$ for MCE irrelevant. We focus on the BLEU score for the remainder of this paper.

As a reminder, to evaluate the BLEU score of a set of N translations e_N^*, against a set of references r_N, we accumulate n-gram precision and closest reference length information for each e_n^* from e_N^* and compute the BLEU score as follows:

$$BLEU(e_N^*, r_N) = \left\{ \sum_{g=1}^{G} w_g \log(p_g) - \max\left(\frac{L_{ref}^*}{L_{sys}} - 1, 0 \right) \right\}$$

$$\text{where}$$

$$p_g = \frac{\sum_{n=1}^{N} \delta_g(e_n^*, r_N)}{\sum_{n=1}^{N} c_g(e_n^*, r_N)}$$

where $\delta_g(e_n^*, r_N)$ counts the number of g-grams matched between the candidate being evaluated and the corresponding reference, and $c_g(e_n^*, r_N)$ counts the number of g-grams suggested in e_n^*. G is the max n-gram size we want to consider.

```
Algorithm 1 identification
Require: N-Best list $E$
Generates: configuration $c$
$c \leftarrow 0$, $c' \leftarrow c$
$b$ = calculateBLEU($c$);
while $c'$ != $c$
    $c=c'$
    for ($n$=1,...$N$)
        foreach (k in NBestList(n))
            $b'$ = calculateBLEU($c'(n) \leftarrow k$)
            if ($b' > b$) { $b=b'$  $c(n) \leftarrow k$ }
        end foreach
    end for
end while
```

4.1. Effect on MMI

For MMI we need to label a "correct" candidate translation for each source sentence from its corresponding N-Best list of candidates. We will call this set of choices a "configuration". Determining a high scoring configuration becomes difficult under the BLEU metric since it operates on a configuration, rather than at the sentence level. Since BLEU scores are not cumulative it is not sufficient to select candidates which are "locally correct" when only compared to other candidates in the same N-Best list. Selecting the true optimal configuration on a training set would require a search through an exponential number of configurations, so we approximate this using an iterative approach. We begin with an initial configuration (usually the top ranked candidate in each N-Best list) and accumulate the relevant statistics for this configuration. Starting with the first N-Best list, consider the impact on the training set score when selecting an alternative translation by subtracting the statistics for the current configuration choice from the accumulated statistics and adding those for the alternative we are considering. Evaluate each alternative in the N-Best list in this fashion, settling on the one that results in the highest training set BLEU score. Repeat this process for the next source sentence, using the locally optimized configuration as a starting point and continue till there are no configuration changes made on the entire corpus. This is effectively a greedy search through the space of configurations and on our training data using 4-Gram BLEU evaluation we see convergence in

2-3 iterations. Pseudo-code detailing this search is shown below. It is important to note that this method will result in a locally optimal configuration. This configuration specifies the “correct” candidate for each N-Best list, and ties represent multiple “correct” hypotheses within an N-Best list. The configuration also provides an estimate of the upper bound for the BLEU score on the training set.

4.2. Effect on MCE

We need to refine our MCE criterion to account for the corpus level BLEU score. Instead of evaluating the error at each sentence, we evaluate the error (negative BLEU score) on the configuration selected by the decision process:

$$F_{MCE} = Error(e_N^*, r_N)$$

5. The Optimization Process

5.1. MMI Optimization

The MMI criterion is optimized using gradient based techniques. The only relevant consideration is over/under flow that might result when computing the $\log \sum_{e_n^k \in E_n} e^{\theta.h(e_n^k, f_n)}$ term. Depending on the sign and magnitude of the feature data term inside the sum might overflow or underflow. We use the following decomposition of $\log \sum_{e_n^k \in E_n} e^{\theta.h(e_n^k, f_n)}$ to prevent this issue:

$$\log \sum_{e_n^k \in E_n} e^{\theta.h(e_n^k, f_n)} = \log\left[e^{\theta.h(e_n^1, f_n)} + e^{\theta.h(e_n^2, f_n)} \cdots e^{\theta.h(e_n^{|E_n|}, f_n)} \right]$$
$$= \log\left[e^{\theta.h(e_n^1, f_n)} . \left(1 + \frac{e^{\theta.h(e_n^2, f_n)}}{e^{\theta.h(e_n^1, f_n)}} \ldots \frac{e^{\theta.h(e_n^{|E_n|}, f_n)}}{e^{\theta.h(e_n^1, f_n)}} \right) \right]$$
$$= \theta.h(e_n^1, f_n) + \log\left[1 + e^{\theta.h(e_n^2, f_n) - \theta.h(e_n^1, f_n)} \ldots e^{\theta.h(e_n^{|E_n|}, f_n) - \theta.h(e_n^1, f_n)} \right]$$

The arguments to the exponential terms are restricted to be the difference in scaled model feature scores between e_n^k and e_n^1, limiting the ability for the exponential calculation to underflow or overflow. This assumes that the scaled model features within a single N-Best list are relatively similar.

5.2. MCE Optimization

The MCE criteria as applied in [Och, 2003] defines a non-smooth error surface in M-dimensional space corresponding to θ. Powell’s method selects a dimension to optimize, while keeping all others fixed, and finds the value of θ_m that minimizes the error on the training set, defining a start point in M-dimensional space to begin optimizing the next dimension. [Och, 2003] proposes an algorithm to significantly reduce the number of evaluations in this greedy search through the M-dimensional space. Each candidate for a given sentence can be represented:

$$e \leftrightarrow \sum_{m=1}^{M} \theta_m * h_m(e,f) = \sum_{m=1, m \neq d}^{M} \theta_m * h_m(e,f) + \theta_d * h_d(e,f)$$
$$a = \sum_{m=1, m \neq d}^{M} \theta_m * h_m(e,f)$$
$$b = h_d(e,f)$$
$$e \leftrightarrow t_{e,f} = a_{e,f} + b_{e,f}\theta_d$$

with the goal of optimizing over dimension d. Each candidate translation in the N-Best list for a particular sentence defines a line in R^2 with respect to d and the total score. The set of candidates in the N-Best list for a given sentence defines a set of lines in R^2 and the decision rule in (2) states that at a given value of $\theta_d = \overset{\wedge}{\theta_d}$, e_n^* is the line with the highest value of the total score. The selection of e_n^* for each sentence at $\overset{\wedge}{\theta_d}$ ultimately determines the error at $\overset{\wedge}{\theta_d}$. Our goal is to find $\theta_d = \theta_d^*$ such that the error is minimized at θ_d^*. Powell’s method would have us evaluating at several values of θ_d to approximate this error surface. As described in [Och, 2003], we can significantly reduce the number of times we need to evaluate this error, by only focusing on values of θ_d that could generate different error values.

We know that the error can only change if we move to a θ_d where the highest line is a different line than before, implying that we only have to evaluate the error at values in between the intersections that line the top surface of the cluster of lines representing the N-Best list for each source sentence. The intersection between any two candidate lines e_1 and e_2 is found at:

$$\overset{\wedge}{\theta_d} = \frac{a_{e,f} - a_{e',f}}{b_{e',f} - b_{e,f}}$$

If we decide to search for these intersection points within a range $[l, r]$ then we can start with the highest line $\arg\max_e t_{e,f}$ at $\theta_d = l$ and search for intersections with all lines that have a steeper slope than this initial line. The intersection point $\theta_d^{\wedge}$ that is closest to l, represents a critical value of θ_d over which the top candidate for this sentence changes. Mark this intersection and repeat the process, looking for intersections on the new candidate line. Repeat until the right boundary is reached.

Each N-Best list generates a set of "critical" values of $\theta_d^{\wedge}$ across which the error contribution from e_n^* might change. We then merge the set of critical values for all sentences by concatenating and sorting them all. In between pairs of boundaries, we know the error must stay constant since the same candidates are selected for all values of θ_d within this boundary. This implies that we only have to evaluate the error within each pair of non-identical boundaries once, to get a complete representation of the error surface with respect to θ_d. When moving onto the next dimension, we set θ_d to the value that generated the lowest error.

Although this implementation significantly reduces the number of times the error needs to be evaluated, we can improve timed performance further with some additional book keeping. Evaluating the training set error at a given θ_d involves evaluating e_n^* for each source sentence and then accumulating statistics from e_N^* to compute the BLEU score.

The selection information has already been computed when evaluating the intersection points. Starting at $\theta_d = l$ we considered intersections with all lines that have a steeper slope. Finding an intersection with a line with a steeper slope implies that the configuration will change over the intersection point. We can define a pair that $\langle \theta_{d,i}, \Delta Error \rangle$ associates the change in error data that occurs when crossing over the i^{th} intersection. For the BLEU score we can store, for each n-gram size, the number of correct and suggested n-grams, as well as the length of the closest reference. Error deltas are then a set of deltas for each relevant statistic.

Figure 1 represents the N-Best list for a single sentence when separating a single dimension. The solid dots on the horizontal axis would be values at which we should evaluate the error. The empty circles are values at which the candidate selection changes.

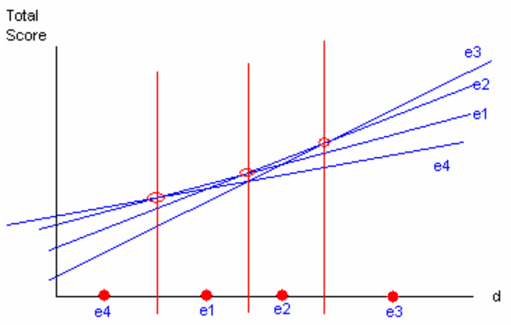

Figure 1. Candidate translation in dimension d, and the critical intersections of one source sentence. Labeling on the d-axis indicates the candidate that the decision rule would choose.

When the $\langle \theta_{d,i}, \Delta Error \rangle$ pairs are merged over all source sentences and sorted according to the intersection, we can simply sum the error deltas as we cross intersection boundaries to track the current value of each statistic. If there are duplicate intersection points in the merged list, we must only consider the error once, since the error data has been summed for all duplicate intersection points (corresponding to changes in the configuration from multiple source sentences). Select θ_d^* as the midpoint of the interval corresponding to the lowest error and continue with the next direction. Termination conditions can be based on the number of iterations or successive reduction of error across iterations.

6. Re-Decoding with θ^*

Decoding the training set with θ^* from MMI or MCE training will generate a new N-Best list for each source sentence. Depending on the nature of the pruning and recombination parameters applied within the decoder, this second list might differ considerably from the N-Best list generated by initial θ, reflecting the local approximation of the candidate space represented by the N-Best list. N-Best lists can be merged across iterations to create a more complete representation of the target sequence search the space. There are two cases to deal with when merging N-Best lists.

Case 1: A new target language candidate sequence is generated – Add the candidate sequence to the merged N-Best list using a dictionary based structure like a trie to conserve space and store the model feature data and relevant BLEU statistics at the leaf node for this target sequence.

Case 2: A target sequence is generated that matches one already in the trie. – Compare the model feature data to those already at the leaf node corresponding to the target sequence and store the additional feature data. This new path corresponds to a different set of decision made by the decoder to generate the same sequence.

Removing duplicate phrase or word translation pairs from the translation models used to decode the training data can reduce the number of candidate data points that have different model feature data but identical target sequences.

It is also important that the choice of θ^* does not affect general parameters of the decoding process. The common Viterbi beam search criterion is particularly affected by changes in θ^*. For example, one variety of beam search restricts the number of partial hypotheses that are expanded over the source sequence by only considering those that have scores within a fixed delta from the top partial hypothesis score at a given source word. The effect of this fixed delta beam changes as the scale of the partial hypothesis scores change. Changes to θ affect the partial hypothesis scores within the decoding process thereby modified the pruning effect of the beam. In addition, if feature scores are dependant on the length of the candidate hypothesis, then the beam has a different effect on sentences of different lengths.

To keep the number of hypotheses considered in each decoding iteration constant, with respect to θ, we use a beam that considers a fixed number of partial hypotheses at each source word.

7. Experimental Results

We evaluate the impact the MMI and MCE criterion have toward improvement on the BLEU score for Chinese to English translation in the newswire domain using data available in the DARPA TIDES evaluations. We use three model features in $P_\theta(e \mid f)$, the log score of a language model built on a 20 million word monolingual corpus, the log score of a translation model, and a sentence length model which simply counts the number of words generated.

We use the decoder and transducers as described in [Vogel, 2003] with the beam modification described earlier. Table 1 details the data characteristics on the small and large track corpora from which transducers are built, and the training and set test, on which θ^* is trained and tested.

Track	#Pairs	Chinese	English
Small	3540	90K	115K
Large	77558	2.46M	2.69M
Training	878	24360	–
Test	919	26223	–

Table 1: Corpus figures indicating no. of sentence pairs and number of Chinese and English word.

On the Small and Large data track we begin with initial scaling factors $\theta = [1,1,1]$, a fixed number of hypotheses beam of size 100, recombination factors that consider the number of words translated; the coverage pattern; the language model state as described in [Vogel, 2003]. We report the improvements in BLEU score due to each method as well as the locally optimal (max) BLEU score, for the small and large track for the training and test data along with the respective generating parameters.

We fix the translation model parameter at 1 to get a better impression of the relative importance of each model. Summary statistics are shown in Tables 2,3 and a graph that details relevant scores across iterations is shown in Figure 1 where TM=Translation Model, LM=Language Model, SL=Sentence Length Model.

	Params(TM, LM,SL)			Max	Train	Test
Base	1.00	1.00	1.00	0.224	0.159	0.163
MCE	1.00	3.72	-0.04	0.261	0.180	0.182
MMI	1.00	4.36	0.59	0.264	0.178	0.180

Table 2: Small track results for the final θ^*

	Params(TM, LM,SL)			Max	Train	Test
Base	1.00	1.00	1.00	0.300	0.243	0.231
MCE	1.00	1.97	-0.31	0.369	0.260	0.251
MMI	1.00	NA	NA	NA	NA	NA

Table 3: Large track results for the final θ^*

The MMI method attempts to separate the top metric scoring hypothesis from competing hypothesis. The top metric scoring hypothesis typically represents translations with lower model costs (higher scores) than the other translations. This effect when considered on large data sets could lead to negative model scores. While good for N-Best list re-ranking this effect would prevent the decoder from exploring the target language search space efficiently. We experienced this issue here, and were unable to generate large track results for the MMI method. We consider this a shortcoming of the MMI method when applied to the translation task using a beam decoder.

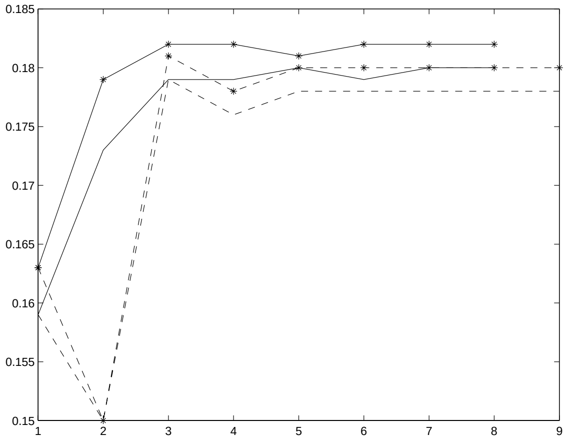

Figure 2 Small track training and test scores, MCE: solid line, MMI dashed line, test score indicated with * on line.

BLEU scores for both models are significantly higher than the initial parameters and the MCE criterion seems to outperform the MMI criterion as expected on the small track and was able to generate large track results, while MMI was unable to. Significance testing shows that the improvements over the baseline are statistically significant over this data [Zhang, Vogel, 2004]. The difference between MMI and MCE in the small track is not statistically significant, (0.005 is the threshold on this dataset).

Small changes in training set scores match quite closely with changes in test set scores implying that the optimal parameters do generalize over test sets. The progress of the MMI method over iterations is significantly more erratic than the MCE method. We believe this comes from the model attempting to further separate already high ranking top candidates, effectively over fitting on the N-Best list, and creating extreme parameter settings that are not effective in re-decoding the training data (an effect which is crippling on the large track). We see evidence of this effect when we look into the parameter setting that caused the plunge in score after the first iteration in the MMI criteria. $\theta = [1,182.2,31.4]$ after the first iteration. While these values have significantly discriminated the top scoring candidate from the alternatives, they are not effective in the translation process.

This problem could be addressed by using a sigmoid smoothing function to limit the effect of severe positive and negative discriminations in the MMI criterion. The MMI criterion is also inherently impaired since a localized selection criterion that must be employed to determine "correct" candidates. By selecting only one candidate, the MMI criterion must discriminate this candidate from all other alternative, regardless of their relative scores. This effect is due to the implied zero-one loss criterion employed in MMI. Alternative approaches could include taking into account relative rank in the N-Best list to weight the contribution to the discriminative criterion.

8. Conclusions

Discriminative training applied to N-Best lists is an effective way to quickly approximate and model the error surface with respect to model parameters. In this work we have compared the formal models as well as empirical results from two classes of discriminative training with the aim of providing a clear framework for reproduction and discussion of results. Our contributions come in the form of detailing the important differences in formalism and practical considerations required to deliver improvements with these methods in the translation domain.

Inspecting the maximum BLEU scores possible on the small and large data tracks showed that while discriminative training has moved us towards these values, we are still significantly far away from making optimal use of the data available in the training corpora. It will be valuable to create methods to determine model specific upper bounds on discriminative training criteria with respect to specific evaluation metrics in the style of [Schlueter, 2001], allowing researchers to focus their efforts towards more accurate estimation of component models or more effective model combination and optimization techniques. We expect to continue our work in showing the relationship between the formal and empirical aspects of discriminative training when applied to statistical machine translation and hope that this work will promote this process in the community.

9. References

Peter F. BROWN, Stephen A. DELLA PIETRA, Vincent J. DELLA PIETRA and Rober L. MERCER. 1993. *The Mathematics of Statistical Machine Translation: Parameter Estimation*, Computational Linguistics vol 19(2) 1993.

George DODDINGTON. 2002. *Automatic evaluation of machine translation quality using n-gram co-occurrence statistics*, In Proc. ARPA Workshop on Human Language Technology, Englewood Cliffs, NJ.

Kishore A. PAPENENI, Salim ROUKOS, Todd WARD, and Wei-Jing ZHU. 2001. *Bleu: a method for automatic evaluation of machine translation,* Technical Report RC 22176, IBM Research Division, Thomas J. Watson Research Center, Yorktown Heights, NY

Franz Josef OCH and Herman NEY. 2002. *Discriminative Training and Maximum Entropy Models for statistical machine translations.* In Proc. of the 40th Annual Meeting of the Association of Computational Linguistics (ACL), Philadelphia, PA, July.

Franz Josef OCH. 2003. *Minimum Error rate in statistical machine translation.* In Proc. of the 41th Annual Meeting of the Association of Computational Linguistics (ACL), Sapporo, Japan.

William H. PRESS, Saul TEUKOLSKY, William T. VETTERLING and Brian P. FLANNERY. 2002. *Numerical Recipes in C++*. Cambridge University Press, Cambridge, UK.

Ralf SCHLUETER and Hermann NEY. 2001. *Model-based MCE bound to the true Bayes error.* IEEE Signal Processing Letters 8(5):131-133, May

M. BEN-BASSAT. 1982. *Use of Distance Features, information measures and error bounds in feature evaluation.* Handbook of Statistics, P.R Krishnaiah and L.N. Kanal, Eds, Amsterdam, The Netherlands: North Holland vol 2 p773-791

B. H. HUANG and S. KATAGIRI. 1992. *Discriminative learning for minimum error classification.* IEEE Trans. Signal Processing vol 40 p3043-3054, Dec

Shankar KUMAR and William BYRNE. 1992. *Discriminative learning for minimum error classification.* In Proc. of the North American Association of Computational Linguistics (HLT-NAACL), Boston, MA.

Libin SHEN, Anoop SARKAR, Franz Josef OCH. 2004. *Discriminative reranking for machine translation.* In Proc. of the North American Association of Computational Linguistics (HLT-NAACL), Boston, MA.

Y. NORMANDIN, R.CARDIN, R. de MORI. 1994. *High performance connected digit recognition using maximum mutual information.* IEEE Transactions on speech and audio processing, vol 2, p299-311, April

Richard ZENS, Hermann NEY. 2003. *Improvements in Phrase-based Statistical Machine Translation.* Human Language Technology Conference, Edmonton, Canada

Stephan VOGEL, Ying ZHANG, Fei HUANG, Alicia TRIBBLE, Ashish VENUGOPAL, Bing ZHAO. 2003. *The CMU Statistical Machine Translation System.* Proc. of The Machine Translation Summit IX

Ying ZHANG, Stephan VOGEL, 2004. *Measuring Confidence Intervals for Machine Translation Evaluation.* TMI, 2004, Baltimore, MD October