# Multi-Metric Optimization Using Ensemble Tuning

**Baskaran Sankaran, Anoop Sarkar**
Simon Fraser University
Burnaby BC. CANADA
{baskaran,anoop}@cs.sfu.ca

**Kevin Duh**
Nara Institute of Science & Technology
Ikoma, Nara. JAPAN
kevinduh@is.naist.jp

## Abstract

This paper examines tuning for statistical machine translation (SMT) with respect to multiple evaluation metrics. We propose several novel methods for tuning towards multiple objectives, including some based on *ensemble decoding* methods. Pareto-optimality is a natural way to think about multi-metric optimization (MMO) and our methods can effectively combine several Pareto-optimal solutions, obviating the need to choose one. Our best performing *ensemble tuning* method is a new algorithm for multi-metric optimization that searches for Pareto-optimal ensemble models. We study the effectiveness of our methods through experiments on multiple as well as single reference(s) datasets. Our experiments show simultaneous gains across several metrics (BLEU, RIBES), without any significant reduction in other metrics. This contrasts the traditional tuning where gains are usually limited to a single metric. Our human evaluation results confirm that in order to produce better MT output, optimizing multiple metrics is better than optimizing only one.

## 1 Introduction

Tuning algorithms are used to find the weights for a statistical machine translation (MT) model by minimizing error with respect to a single MT evaluation metric. The tuning process improves the performance of an SMT system as measured by this metric; with BLEU (Papineni et al., 2002) being the most popular choice. Minimum error-rate training (MERT) (Och, 2003) was the first approach in MT to directly optimize an evaluation metric. Several alternatives now exist: MIRA (Watanabe et al., 2007; Chiang et al., 2008), PRO (Hopkins and May, 2011), linear regression (Bazrafshan et al., 2012) and ORO (Watanabe, 2012) among others.

However these approaches optimize towards the best score as reported by a single evaluation metric. MT system developers typically use BLEU and ignore all the other metrics. This is done despite the fact that other metrics model wide-ranging aspects of translation: from measuring the translation edit rate (TER) in matching a translation output to a human reference (Snover et al., 2006), to capturing lexical choices in translation as in METEOR (Lavie and Denkowski, 2009) to modelling semantic similarity through textual entailment (Padó et al., 2009) to RIBES, an evaluation metric that pays attention to long-distance reordering (Isozaki et al., 2010). While some of these metrics such as TER, METEOR are gaining prominence, BLEU enjoys the status of being the *de facto* standard tuning metric as it is often claimed and sometimes observed that optimizing with BLEU produces better translations than other metrics (Callison-Burch et al., 2011).

The gains obtained by the MT system tuned on a particular metric do not improve performance as measured under other metrics (Cer et al., 2010), suggesting that over-fitting to a specific metric might happen without improvements in translation quality. In this paper we propose a new tuning framework for jointly optimizing multiple evaluation metrics.

Pareto-optimality is a natural way to think about multi-metric optimization and multi-metric optimization (MMO) was recently explored using the notion of Pareto optimality in the Pareto-based Multi-objective Optimization (PMO) approach (Duh et al., 2012). PMO provides several equivalent solutions (parameter weights) having different trade-offs between the different MT metrics. In (Duh et al., 2012) the choice of which option to use rests with the MT system developer and in that sense their approach is an *a posteriori* method to specify the preference (Marler and Arora, 2004).

In contrast to this, our tuning framework provides a principled way of using the Pareto optimal options using *ensemble* decoding (Razmara et al., 2012). We also introduce a novel method of *ensemble tuning* for jointly tuning multiple MT evaluation metrics and further combine this with the PMO ap-



*Proceedings of NAACL-HLT 2013*, pages 947–957,
Atlanta, Georgia, 9–14 June 2013. 

proach (Duh et al., 2012). We also introduce three other approaches for multi-metric tuning and compare their performance to the ensemble tuning. Our experiments yield the highest metric scores across many different metrics (that are being optimized), something that has not been possible until now.

Our ensemble tuning method over multiple metrics produced superior translations than single metric tuning as measured by a post-editing task. HTER (Snover et al., 2006) scores in our human evaluation confirm that multi-metric optimization can lead to better MT output.

## 2 Related Work

In grammar induction and parsing (Spitkovsky et al., 2011; Hall et al., 2011; Auli and Lopez, 2011) have proposed multi-objective methods based on round-robin iteration of single objective optimizations.

Research in SMT parameter tuning has seen a surge of interest recently, including online/batch learning (Watanabe, 2012; Cherry and Foster, 2012), large-scale training (Simianer et al., 2012; He and Deng, 2012), and new discriminative objectives (Gimpel and Smith, 2012; Zheng et al., 2012; Bazrafshan et al., 2012). However, few works have investigated the multi-metric tuning problem in depth. Linear combination of BLEU and TER is reported in (Zaidan, 2009; Dyer et al., 2009; Servan and Schwenk, 2011); an alternative is to optimize on BLEU with MERT while enforcing that TER does not degrade per iteration (He and Way, 2009). Studies on metric tunability (Liu et al., 2011; Callison-Burch et al., 2011; Chen et al., 2012) have found that the metric used for evaluation may not be the best metric used for tuning. For instance, (Mauser et al., 2008; Cer et al., 2010) report that tuning on linear combinations of BLEU-TER is more robust than a single metric like WER.

The approach in (Devlin and Matsoukas, 2012) modifies the optimization function to include traits such as output length so that the hypotheses produced by the decoder have maximal score according to one metric (BLEU) but are subject to an output length constraint, e.g. that the output is 5% shorter. This is done by rescoring an N-best list (forest) for the metric combined with each trait condition and then the different trait hypothesis are combined using a system combination step. The traits are independent of the reference (while tuning). In contrast, our method is able to combine multiple metrics (each of which compares to the reference) during the tuning step and we do not depend on N-best list (or forest) rescoring or system combination.

Duh et. al. (2012) proposed a Pareto-based approach to SMT multi-metric tuning, where the linear combination weights do not need to be known in advance. This is advantageous because the optimal weighting may not be known in advance. However, the notion of Pareto optimality implies that multiple "best" solutions may exist, so the MT system developer may be forced to make a choice after tuning.

These approaches require the MT system developer to make a choice either before tuning (e.g. in terms of linear combination weights) or afterwards (e.g. the Pareto approach). Our method here is different in that we do not require any choice. We use *ensemble decoding* (Razmara et al., 2012) (see sec 3) to combine the different solutions resulting from the multi-metric optimization, providing an elegant solution for deployment. We extend this idea further and introduce *ensemble tuning*, where the metrics have separate set of weights. The tuning process alternates between ensemble decoding and the update step where the weights for each metric are optimized separately followed by joint update of metric (meta) weights.

## 3 Ensemble Decoding

We now briefly review ensemble decoding (Razmara et al., 2012) which is used as a component in the algorithms we present. The prevalent model of statistical MT is a log-linear framework using a vector of feature functions $\phi$:

$$p(e|f) \propto \exp\Big(w \cdot \phi\Big) \tag{1}$$

The idea of ensemble decoding is to combine several models dynamically at decode time. Given multiple models, the scores are combined for each partial hypothesis across the different models during decoding using a user-defined mixture operation $\otimes$.

$$p(e|f) \propto \exp\Big(w_1 \cdot \phi_1 \otimes w_2 \cdot \phi_2 \otimes \dots\Big) \tag{2}$$

(Razmara et al., 2012) propose several mixture operations, such as *log-wsum* (simple linear mixture), *wsum* (log-linear mixture) and *max* (choose lo-

cally best model) among others. The different mixture operations allows the user to encode the beliefs about the relative strengths of the models. It has been applied successfully for domain adaptation setting and shown to perform better approaches that pre-compute linear mixtures of different models.

## 4 Multi-Metric Optimization

In statistical MT, the multi-metric optimization problem can be expressed as:

$$w^* = \arg\max_w g\Big( [M_1(H), \ldots, M_k(H)] \Big) \quad (3)$$
$$\text{where } H = \mathcal{N}(\mathbf{f}; w)$$

where $\mathcal{N}(\mathbf{f}; w)$ is the decoding function generating a set of candidate hypotheses $H$ based on the model parameters $w$, for the source sentences $\mathbf{f}$. For each source sentence $f_i \in \mathbf{f}$ there is a set of candidate hypotheses $\{h_i\} \in H$. The goal of the optimization is to find the weights that maximize the function $g(.)$ parameterized by different evaluation metrics $M_1, \ldots, M_k$.

For the Pareto-optimal based approach such as PMO (Duh et al., 2012), we can replace $g(\cdot)$ above with $g_{\text{PMO}}(\cdot)$ which returns the points in the Pareto frontier. Alternately a weighted averaging function $g_{\text{wavg}}(\cdot)$ would result in a linear combination of the metrics being considered, where the tuning method would maximize the joint metric. This is similar to the (TER-BLEU)/2 optimization (Cer et al., 2010; Servan and Schwenk, 2011).

We introduce four methods based on the above formulation and each method uses a different type of $g(\cdot)$ function for combining different metrics and we compare experimentally with existing methods.

### 4.1 PMO Ensemble

PMO (Duh et al., 2012) seeks to maximize the number of points in the Pareto frontier of the metrics considered. The inner routine of the PMO-PRO tuning is described in Algorithm 1. This routine is contained within an outer loop that iterates for a fixed number iterations of decoding the tuning set and optimizing the weights.

The tuning process with PMO-PRO is independently repeated with different set of weights for metrics[1] yielding a set of equivalent solutions

**Algorithm 1** PMO-PRO (*Inner routine for tuning*)

**Input:** Hypotheses H = $\mathcal{N}(\mathbf{f}; w)$; Weights $w$
Initialize $\mathcal{T} = \{\}$
**for** each f in tuning set **f do**
  $\{h\} = H(f)$
  $\{M(\{h\})\}$ = ComputeMetricScore($\{h\}, \hat{e}$)
  $\{\mathcal{F}\}$ = FindParetoFrontier($\{M(\{h\})\}$)
  **for** each h in $\{h\}$ **do**
    **if** $h \in \mathcal{F}$ **then** add $(1, h)$ to $\mathcal{T}$
    **else** add $(\ell, h)$ to $\mathcal{T}$  (see footnote 1)
$w^p \leftarrow$ PRO($\mathcal{T}$) (optimize using PRO)
**Output:** Pareto-optimal weights $w^p$

$\{p_{s_1}, \ldots, p_{s_n}\}$ which are points on the Pareto frontier. The user then chooses one solution by making a trade-off between the performance gains across different metrics. However, as noted earlier this *a posteriori* choice ignores other solutions that are indistinguishable from the chosen one.

We alleviate this by complementing PMO with ensemble decoding, which we call *PMO ensemble*, in which each point in the Pareto solution is a distinct component in the ensemble decoder. This idea can also be used in other MMO approaches such as linear combination of metrics ($g_{\text{wavg}}(.)$) mentioned above. In this view, PMO ensemble is a special case of *ensemble combination*, where the decoding is performed by an ensemble of optimal solutions.

The ensemble combination model introduces new hyperparameters $\boldsymbol{\beta}$ that are the weights of the ensemble components (meta weights). These ensemble weights could set to be uniform in a naïve implementation. Or the user can encode her beliefs or expectations about the individual solutions $\{p_{s_1}, \ldots, p_{s_n}\}$ to set the ensemble weights (based on the relative importance of the components). Finally, one could also include a meta-level tuning step to set the weights $\boldsymbol{\beta}$.

The PMO ensemble approach is graphically illustrated in Figure 1; we will also refer to this figure while discussing other methods.[2] The orig-

[1]For example Duh et al. (2012) use five different weight settings for metrics $(M_1, M_2)$, viz. $(0.0, 1.0)$, $(0.3, 0.7)$, $(0.5, 0.5)$, $(0.7, 0.3)$ and $(1.0, 0.0)$. They combine the metric weights $q_i$ with the sentence-level metric scores $M_i$ as $\ell = \left(\sum_k q_k M_k\right)/k$ where $\ell$ is the target value for negative examples (the *else* line in Alg 1) in the optimization step.

[2]The illustration is based on two metrics, metric-1 and metric-2, but could be applied to any number of metrics. Without loss of generality we assume accuracy metrics, i.e. higher

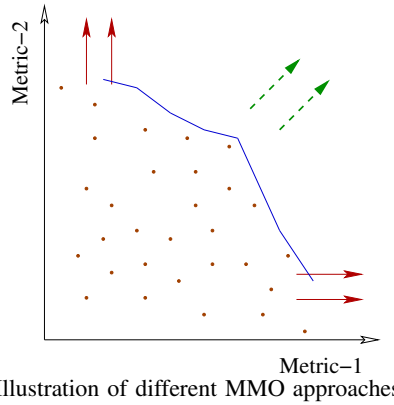


Figure 1: Illustration of different MMO approaches involving two metrics. Solid (red) arrows indicate optimizing two metrics independently and the dashed (green) arrow optimize them jointly. The Pareto frontier is indicated by the curve.

inal PMO-PRO seeks to maximize the points on the Pareto frontier (blue curve in the figure) leading to Pareto-optimal solutions. On the other hand, the PMO ensemble combines the different Pareto-optimal solutions and potentially moving in the direction of dashed (green) arrows to some point that has higher score in either or both dimensions.

### 4.2 Lateen MMO

*Lateen* EM has been proposed as a way of jointly optimizing multiple objectives in the context of dependency parsing (Spitkovsky et al., 2011). It uses a secondary hard EM objective to move away, when the primary soft EM objective gets stuck in a local optima. The course correction could be performed under different conditions leading to variations that are based on when and how often to shift from one objective function to another during optimization.

The lateen technique can be applied to the multi-metric optimization in SMT by treating the different metrics as different objective functions. While the several lateen variants are also applicable for our task, our objective here is to improve performance across the different metrics (being optimized). Thus, we restrict ourselves to the style where the search alternates between the metrics (in round-robin fashion) at each iteration. Since the notion of convergence is unclear in lateen setting, we stop after a fixed number of iterations optimizing the tuning set. In terms of Figure 1, lateen MMO corresponds to alternately maximizing the metrics along two dimensions as depicted by the solid arrows.

By the very nature of lateen-alternation, the weights obtained at each iteration are likely to be best for the metric that was optimized in that iteration. Thus, one could use weights from the last $k$ iterations (for lateen-tuning with as many metrics) and then decode the test set with an ensemble of these weights as in PMO ensemble. However in practice we find the weights to converge and we simply use the weights from the final iteration to decode the test set in our lateen experiments.

### 4.3 Union of Metrics

At each iteration lateen MMO excludes all but one metric for optimization. An alternative would be to consider all the metrics at each iteration so that the optimizer could try to optimize them jointly. This has been the general motivation for considering the linear combination of metrics (Cer et al., 2010; Servan and Schwenk, 2011) resulting in a joint metric, which is then optimized.

However due to the scaling differences between the scores of different metrics, the linear combination might completely suppress the metric having scores in the lower-range. As an example, the RIBES scores that are typically in the high 0.7-0.8 range, dominate the BLEU scores that is typically around 0.3. While the weighted linear combination tries to address this imbalance, they introduce additional parameters that are manually fixed and not separately tuned.

We avoid this linear combination pitfall by taking the union of the metrics under which we consider the union of training examples from all metrics and optimize them jointly. Mathematically,

$$w^* = \arg\max_{w} \; g(M_1(H)) \cup \ldots \cup g(M_k(H)) \quad (4)$$

Most of the optimization approaches involve two phases: i) select positive and negative examples and ii) optimize parameters to favour positive examples while penalizing negative ones. In the *union* approach, we independently generate positive and negative sets of examples for all the metrics and take their union. The optimizer now seeks to move towards positive examples from all metrics, while penalizing others.

This is similar to the PMO-PRO approach except that here the optimizer tries to simultaneously maximize the number of high scoring points across all

metric score is better.

metrics. Thus, instead of the entire Pareto frontier curve in Figure 1, the union approach optimizes the two dimensions simultaneously in each iteration.

## 5 Ensemble Tuning

These methods, even though novel, under utilize the power of ensembles as they combine the solution only at the end of the tuning process. We would prefer to tightly integrate the idea of ensembles into the tuning. We thus extend the ensemble decoding to *ensemble tuning*. The feature weights are replicated separately for each evaluation metric, which are treated as components in the ensemble decoding and tuned independently in the optimization step. Initially the ensemble decoder decodes a devset using a weighted ensemble to produce a single N-best list. For the optimization, we employ a two-step approach of optimizing the feature weights (of each ensemble component) followed by a step for tuning the meta (component) weights. The optimized weights are then used for decoding the devset in the next iteration and the process is repeated for a fixed number of iterations.

Modifying the MMO representation in Equation 3, we formulate *ensemble tuning* as:

$$H_{ens} = \mathcal{N}_{ens}\Big(\mathbf{f}; \{w_M\}; \otimes; \boldsymbol{\lambda}\Big) \quad (5)$$

$$\mathbf{w}^* = \Big\{ \arg\max_{w_{M_i}} H_{ens} \mid 1 \leq i \leq k \Big\} \quad (6)$$

$$\boldsymbol{\lambda} = \arg\max_{\lambda} g\left(\{M_i(H_{ens}) | 1 \leq i \leq k\}; \mathbf{w}^*\right) \quad (7)$$

Here the ensemble decoder function $\mathcal{N}_{ens}(.)$ is parameterized by an ensemble of weights $w_{M_1}, \ldots, w_{M_k}$ (denoted as $\{w_M\}$ in Eq 5) for each metric and a mixture operation ($\otimes$). $\boldsymbol{\lambda}$ represents the weights of the ensemble components.

Pseudo-code for ensemble tuning is shown in Algorithm 2. In the beginning of each iteration (line 2), the tuning process ensemble decodes (line 4) the tuning set using the weights obtained from the previous iteration. Equation 5 gives the detailed expression for the ensemble decoding, where $H_{ens}$ denotes the N-best list generated by the ensemble decoder.

The method now uses a dual tuning strategy involving two phases to optimize the weights. In the first step it optimizes each of the $k$ metrics independently (lines 6-7) along its respective dimension in the multi-metric space (as shown by the solid arrows along the two axes in Figure 1). This yields a new set of weights $\mathbf{w}^*$ for the features in each metric.

**Algorithm 2** Ensemble Tuning Algorithm

1: **Input:** Tuning set $\mathbf{f}$,
   Metrics $M_1, \ldots, M_k$ (ensemble components)
   Initial weights $\{w_M\} \leftarrow w_{M_1}, \ldots w_{M_k}$ and
   Component (meta) weights $\boldsymbol{\lambda}$
2: **for** $j = 1, \ldots$ **do**
3: $\quad \{w_M^{(j)}\} \leftarrow \{w_M\}$
4: $\quad$ *Ensemble* decode the tuning set
   $\quad\quad H_{ens} = \mathcal{N}_{ens}(\mathbf{f}; \{w_M^{(j)}\}; \otimes; \boldsymbol{\lambda})$
5: $\quad \{w_M\} = \{\}$
6: $\quad$ **for** each metric $M_i \in \{M\}$ **do**
7: $\quad\quad w^*_{M_i} \leftarrow$ PRO$(H_{ens}, w_{M_i})$ (use PRO)
8: $\quad\quad$ Add $w^*_{M_i}$ to $\{w_M\}$
9: $\quad \boldsymbol{\lambda} \leftarrow$ PMO-PRO$(H_{ens}, \{w_M\})$ (Alg 1)
10: **Output:** Optimal weights $\{w^M\}$ and $\boldsymbol{\lambda}$

The second tuning step (line 9) then optimizes the meta weights ($\boldsymbol{\lambda}$) so as to maximize the multi-metric objective along the joint $k$-dimensional space as shown in Equation 7. This is illustrated by the dashed arrows in the Figure 1. While $g(.)$ could be any function that combines multiple metrics, we use the PMO-PRO algorithm (Alg. 1) for this step.

The main difference between *ensemble tuning* and *PMO ensemble* is that the former is an ensemble model over metrics and the latter is an ensemble model over Pareto solutions. Additionally, PMO ensemble uses the notion of ensembles only for the final decoding after tuning has completed.

### 5.1 Implementation Notes

All the proposed methods fit naturally within the usual SMT tuning framework. However, some changes are required in the decoder to support ensemble decoding and in the tuning scripts for optimizing with multiple metrics. For ensemble decoding, the decoder should be able to use multiple weight vectors and dynamically combine them according to some desired mixture operation. Note that, unlike Razmara et al. (2012), our approach uses just one model but has different weight vectors for each metric and the required decoder modifications are simpler than full ensemble decoding.

While any of the mixture operations proposed by Razmara et al. (2012) could be used, in this pa-

per we use *log-wsum* – the linear combination of the ensemble components and *log-wmax* – the combination that prefers the locally best component. These are simpler to implement and also performed competitively in their domain adaptation experiments. Unless explicitly noted otherwise, the results presented in Section 6 are based on linear mixture operation log-wsum, which empirically performed better than the log-wmax for ensemble tuning.

## 6 Experiments

We evaluate the different methods on Arabic-English translation in single as well as multiple references scenario. Corpus statistics are shown in Table 1. For all the experiments in this paper, we use Kriya, our in-house Hierarchical phrase-based (Chiang, 2007) (Hiero) system, and integrated the required changes for ensemble decoding. Kriya performs comparably to the state of the art in phrase-based and hierarchical phrase-based translation over a wide variety of language pairs and data sets (Sankaran et al., 2012).

We use PRO (Hopkins and May, 2011) for optimizing the feature weights and PMO-PRO (Duh et al., 2012) for optimizing meta weights, wherever applicable. In both cases, we use SVM-Rank (Joachims, 2006) as the optimizer.

We used the default parameter settings for different MT tuning metrics. For METEOR, we tried both METEOR-tune and METEOR-hter settings and found the latter to perform better in BLEU and TER scores, even though the former was marginally better in METEOR[3] and RIBES scores. We observed the margin of loss in BLEU and TER to outweigh the gains in METEOR and RIBES and we chose METEOR-hter setting for both optimization and evaluation of all our experiments.

### 6.1 Evaluation on Tuning Set

Unlike conventional tuning methods, PMO (Duh et al., 2012) was originally evaluated on the tuning set to avoid potential mismatch with the test set. In order to ensure robustness of evaluation, they re-decode the devset using the optimal weights from the last tuning iteration and report the scores on 1-best candidates.

| Corpus | Training size | Tuning/ test set |
|---|---|---|
| ISI corpus | 1.1 M | 1664/ 1313 (MTA) |
| | | 1982/ 987 (ISI) |

Table 1: Corpus Statistics (# of sentences) for Arabic-English. MTA (4-refs) and ISI (1-ref).

We follow the same strategy and compare our PMO-ensemble approach with PMO-PRO (denoted P) and a linear combination[4] (denoted L) baseline. Similar to Duh et al. (2012), we use five different BLEU:RIBES weight settings, viz. $(0.0, 1.0)$, $(0.3, 0.7)$, $(0.5, 0.5)$, $(0.7, 0.3)$ and $(1.0, 0.0)$, marked L1 through L5 or P1 through P5. The Pareto frontier is then computed from 80 points (5 runs and 15 iterations per run) on the devset.

Figure 2(a) shows the Pareto frontier of L and P baselines using BLEU and RIBES as two metrics. The frontier of the P dominates that of L for most part showing that the PMO approach benefits from picking Pareto points during the optimization.

We use the PMO-ensemble approach to combine the optimized weights from the 5 tuning runs and re-decode the devset employing ensemble decoding. This yields the points *LEns* and *PEns* in the plot, which obtain better scores than most of the individual runs of L and P. This ensemble approach of combining the final weights also generalizes to the unseen test set as we show later.

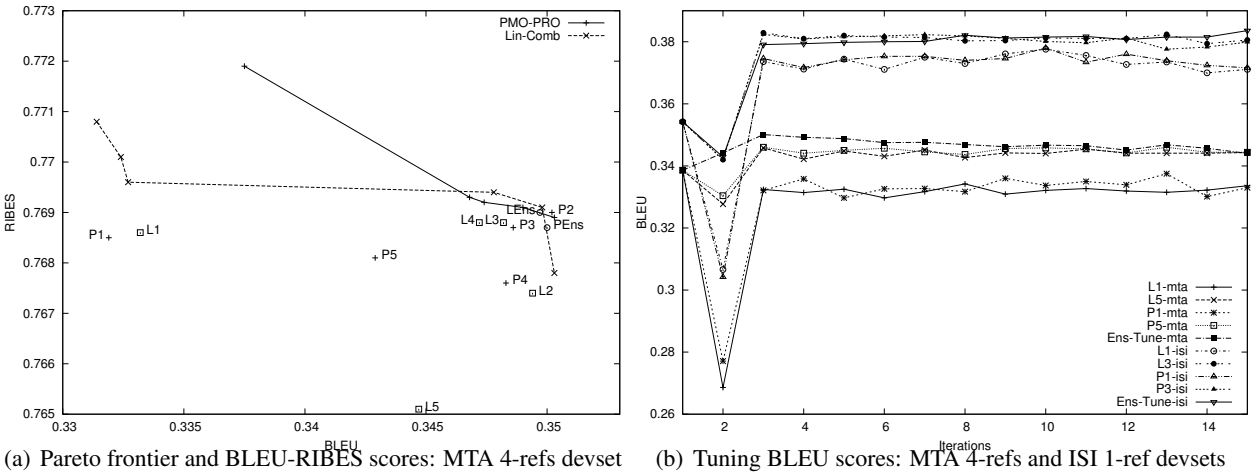
Figure 2(b) plots the change in BLEU during tuning in the multiple references and the single reference scenarios. We show for each baseline method L and P, plots for two different weight settings that obtain high BLEU and RIBES scores. In both datasets, our ensemble tuning approach dominates the curves of the (L and P) baselines. In summary, these results confirm that the ensemble approach achieves results that are competitive with previous MMO methods on the devset Pareto curve. We now provide a more comprehensive evaluation on the test set.

### 6.2 Evaluation on Test Set

This section contains multi-metric optimization results on the unseen test sets, one test set has multiple references and the other has a single-reference.

[3]This behaviour was also noted by Denkowski and Lavie (2011) in their analysis of Urdu-English system for *tunable metrics* task in WMT11.

[4]Linear combination is a generalized version of the combined (TER-BLEU)/2 metric and its variants.




(a) Pareto frontier and BLEU-RIBES scores: MTA 4-refs devset (b) Tuning BLEU scores: MTA 4-refs and ISI 1-ref devsets

Figure 2: Devset (redecode): Comparison of Lin-comb (L) and PMO-PRO (P) with Ensemble decoding (Lens and PEns) and Ensemble tuning (Ens-Tune)

We plot BLEU scores against other metrics (RIBES, METEOR and TER) and this allows us to compare the performance of each metric relative to the de-facto standard BLEU metric.

Baseline points are identified by single letters B for BLEU, T for TER, etc. and the baseline (single-metric optimized) score for each metric is indicated by a dashed line on the corresponding axis. MMO points use a series of single letters referring to the metrics used, e.g. BT for BLEU-TER. The union of metrics method is identified with the suffix 'J' and lateen method with suffix 'L' (thus BT-L refers to the *lateen* tuning with BLEU-TER). MMO points without any suffix use the ensemble tuning approach.

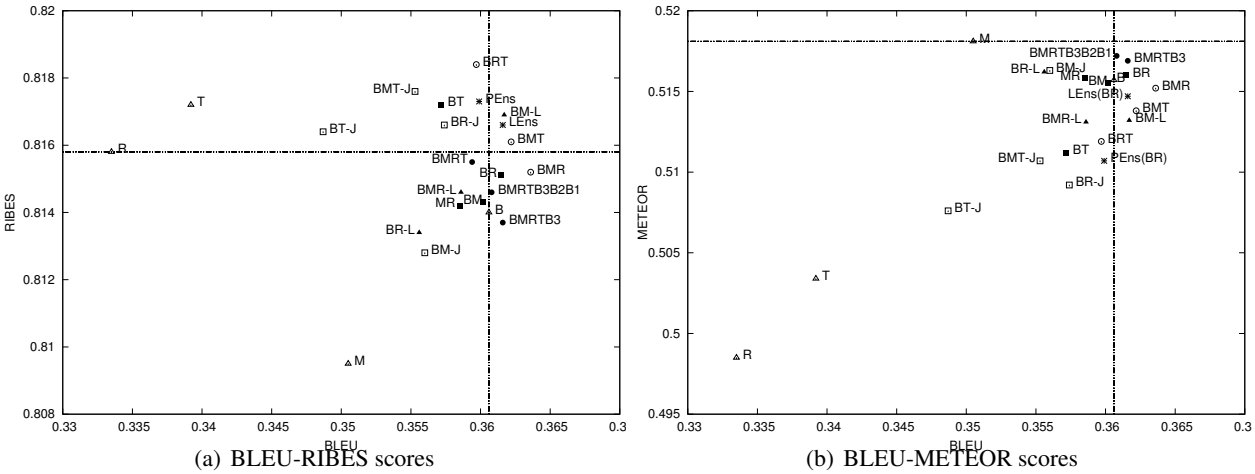
Figures 3 and 4(a) plot the scores for the MTA test set with 4-references. We see noticeable and some statistically significant improvements in BLEU and RIBES (see Table 2 for BLEU improvements). All our MMO approaches, except for the union method, show gains on both BLEU and RIBES axes. Figures 3(b) and 4(a) show that none of the proposed methods managed to improve the baseline scores for METEOR and TER. However, several of our ensemble tuning combinations work well for both METEOR (BR, BMRTB3, etc.) and TER (BMRT and BRT) in that they improved or were close to the baseline scores in either dimension. We again see in these figures that the MMO approaches can improve the BLEU-only tuning by 0.3 BLEU points, without much drop in other metrics. This is in tune with the finding that BLEU could be tuned easily (Callison-Burch et al., 2011) and also explains why it remains a popular choice for optimizing SMT systems.

| Approach and Tuning Metric(s) | BLEU | |
|---|---|---|
| | MTA | ISI |
| Single Objective Baselines | | |
| BLEU | 36.06 | 37.20 |
| METEOR | 35.05 | 36.91 |
| RIBES | 33.35 | 36.60 |
| TER | 33.92 | 35.85 |
| Ensemble Tuning: 2 Metrics | | |
| B-M | 36.02 | ***37.26*** |
| B-R | **36.15** | ***37.37*** |
| B-T | 35.72 | 36.31 |
| Ensemble Tuning: 3 Metrics | | |
| B-M-R | ***36.36*** | ***37.37*** |
| B-M-T | **36.22** | 36.89 |
| B-R-T | 35.97 | 36.72 |
| Ensemble Tuning: $>$ 3 Metrics | | |
| B-M-R-T | 35.94 | 36.84 |
| B-M-R-T-B3 | ***36.16*** | **37.12** |
| B-M-R-T-B3-B2-B1 | ***36.08*** | ***37.24*** |

Table 2: BLEU Scores on MTA (4 refs) and ISI (1 ref) test sets using the standard *mteval* script. Boldface scores indicate scores that are comparable to or better than the baseline BLEU-only tuning. *Italicized* scores indicate statistically significant differences at $p$-value 0.05 computed with bootstrap significance test.

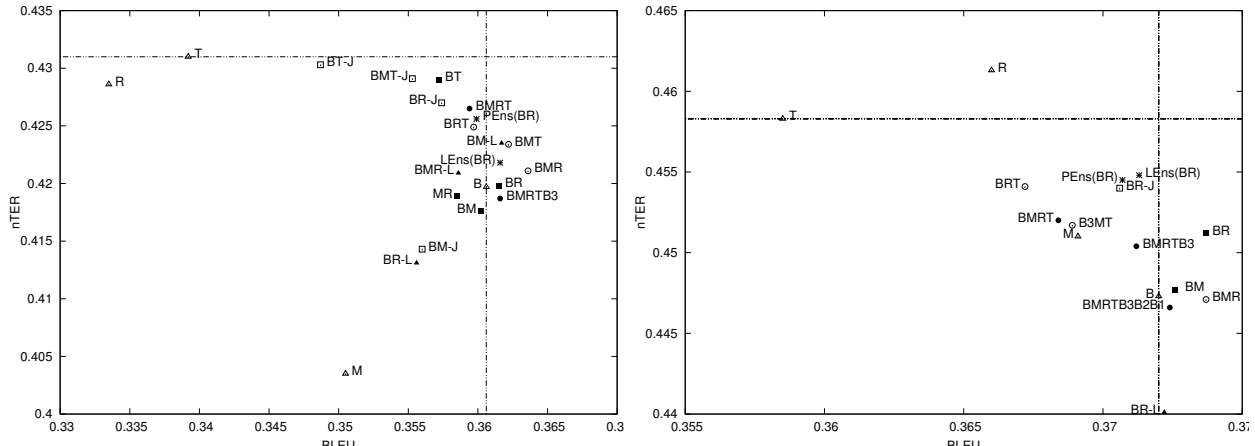
Among the different MMO methods the ensemble tuning performs better than lateen or union approaches. In terms of the number of metrics being optimized jointly, we see substantial gains when using a small number (typically 2 or 3) of metrics. Results seem to suffer beyond this number; probably because there might not be a space that contain solution(s) optimal for *all* the metrics that are jointly optimized.

We hypothesize that each metric correlates well




(a) BLEU-RIBES scores (b) BLEU-METEOR scores

Figure 3: MTA 4-refs testset: Comparison of different MMO approaches. The dashed lines correspond to baseline scores tuned on the respective metrics in the axes. The union of metrics method is identified with the suffix J and lateen with suffix L.




(a) MTA (4-refs) (b) ISI (1-ref)

Figure 4: BLEU-TER scores: Comparison of different MMO approaches. We plot nTER (1-TER) scores for easy reading of the plots. The dashed lines correspond to baseline scores tuned on the respective metrics in the axes.

(in a looser sense) with few others, but not all. For example, union optimizations BR-J and BMT-J perform close to or better than RIBES and TER baselines, but get very poor score in METEOR. On the other hand BM-J is close to the METEOR baseline, while doing poorly on the RIBES and TER. This behaviour is also evident from the single-metric baselines, where R and T-only settings are clearly distinguished from the M-only system. It is not clear if such distinct classes of metrics could be bridged by some optimal solution and the *metric dichotomy* requires further study as this is key to practical multi-metric tuning in SMT.

The lateen and union approaches appear to be very sensitive to the number of metrics and they generally perform well for two metrics case and show degradation for more metrics. Unlike other approaches, the union approach failed to improve over the baseline BLEU and this could be attributed to the conflict of interest among the metrics, while choosing example points for the optimization step. The positive example preferred by a particular metric could be a negative example for the other metric. This would only confuse the optimizer resulting in poor solutions. Our future line of work would be to study the effect of avoiding such of conflicting examples in the union approach.

For the single-reference (ISI) dataset, we only plot the BLEU-TER case in Figure 4(b) due to lack of space. The results are similar to the multiple references set indicating that MMO approaches are equally effective for single references[5]. Table 2

[5]One could argue that MMO methods require multiple references since each metric might be picking out a different ref-

| Metric | Single-metric Tuning | | Ensemble Tuning |
|---|---|---|---|
| | B-only | M-only | B-M-R |
| BLEU | 37.89 | 37.18 | ***39.01*** |
| HBLEU | 51.93 | **53.59** | 53.14 |
| METEOR | 61.31 | 61.56 | **61.68** |
| HMETEOR | 72.35 | 72.39 | **72.74** |
| TER | 0.520 | 0.532 | ***0.516*** |
| HTER | 0.361 | 0.370 | **0.346** |

Table 3: Post-editing Human Evaluation: Regular (untargeted) and human-targeted scores. Human targeted scores are computed against the post-edited reference and regular scores are computed with the original references. **Best** scores are in boldface and *statistically significant* ones (at $p = 0.05$) are italicized.

shows the BLEU scores for our ensemble tuning method (for various combinations) and we again see improvements over the baseline BLEU-only tuning.

### 6.3 Human Evaluation

So far we have shown that multi-metric optimization can improve over single-metric tuning on a single metric like BLEU and we have shown that our methods find a tuned model that performs well with respect to multiple metrics. Is the output that scores higher on multiple metrics actually a better translation? To verify this, we conducted a post-editing human evaluation experiment. We compared our ensemble tuning approach involving BLEU, METEOR and RIBES (B-M-R) with systems optimized for BLEU (B-only) and METEOR (M-only).

We selected 100 random sentences (that are at least 15 words long) from the Arabic-English MTA (4 references) test set and translated them using the three systems (two single metric systems and BMR ensemble tuning). We shuffled the resulting translations and split them into 3 sets such that each set has equal number of the translations from three systems. The translations were edited by three human annotators in a post-editing setup, where the goal was to edit the translations to make them as close to the references as possible, using the Post-Editing Tool: PET (Aziz et al., 2012). The annotators were not Arabic-literate and relied only on the reference translations during post-editing. The identifiers that link each translation to the system that generated it are removed to avoid annotator bias.

In the end we collated post-edited translations for each system and then computed the system-level human-targeted (HBLEU, HMETEOR, HTER) scores, by using respective post-edited translations as the reference. First comparing the HTER (Snover et al., 2006) scores shown in Table 3, we see that the single-metric system optimized for METEOR performs slightly worse than the one optimized for BLEU, despite using METEOR-hter version (Denkowski and Lavie, 2011). Ensemble tuning-based system optimized for three metrics (B-M-R) improves HTER by 4% and 6.3% over BLEU and METEOR optimized systems respectively.

The single-metric system tuned with M-only setting scores high on HBLEU, closely followed by the ensemble system. We believe this to be caused by chance rather than any systematic gains by the M-only tuning; the ensemble system scores high on HMETEOR compared to the M-only system. While HTER captures the edit distance to the targeted reference, HMETEOR and HBLEU metrics capture missing content words or synonyms by exploiting $n$-grams and paraphrase matching.

We also computed the regular variants (BLEU, METEOR and TER), which are scored against original references. The ensemble system outperformed the single-metric systems in all the three metrics. The improvements were also statistically significant at $p$-value of $0.05$ for BLEU and TER.

## 7 Conclusion

We propose and present a comprehensive study of several multi-metric optimization (MMO) methods in SMT. First, by exploiting the idea of ensemble decoding (Razmara et al., 2012), we propose an effective way to combine multiple Pareto-optimal model weights from previous MMO methods (e.g. Duh et al. (2012)), obviating the need for manually trading off among metrics. We also proposed two new variants: lateen-style MMO and union of metrics.

We also extended ensemble decoding to a new tuning algorithm called *ensemble tuning*. This method demonstrates statistically significant gains for BLEU and RIBES with modest reduction in METEOR and TER. Further, in our human evaluation, ensemble tuning obtains the best HTER among competing baselines, confirming that optimizing on multiple metrics produces human-preferred translations compared to the conventional optimization approach involving a single metric.

erence sentence. Our experiment shows that even with a single reference MMO methods can work.

## References


Michael Auli and Adam Lopez. 2011. Training a log-linear parser with loss functions via softmax-margin. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 333–343, Edinburgh, Scotland, UK., July. Association for Computational Linguistics.

Wilker Aziz, Sheila Castilho Monteiro de Sousa, and Lucia Specia. 2012. PET: a tool for post-editing and assessing machine translation. In *Proceedings of the Eight International Conference on Language Resources and Evaluation (LREC'12)*, Istanbul, Turkey, may. European Language Resources Association (ELRA).

Marzieh Bazrafshan, Tagyoung Chung, and Daniel Gildea. 2012. Tuning as linear regression. In *Proceedings of the 2012 Conference of the North American Chapter of the ACL: Human Language Technologies*, pages 543–547, Montréal, Canada. ACL.

Chris Callison-Burch, Philipp Koehn, Christof Monz, and Omar Zaidan. 2011. Findings of the 2011 workshop on statistical machine translation. In *Proceedings of the Sixth Workshop on Statistical Machine Translation*, pages 22–64, Edinburgh, Scotland, July. ACL.

Daniel Cer, Christopher D. Manning, and Daniel Jurafsky. 2010. The best lexical metric for phrase-based statistical mt system optimization. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the ACL*, pages 555–563. ACL.

Boxing Chen, Roland Kuhn, and Samuel Larkin. 2012. Port: a precision-order-recall mt evaluation metric for tuning. In *Proceedings of the 50th Annual Meeting of the ACL (Volume 1: Long Papers)*, pages 930–939, Jeju Island, Korea. ACL.

Colin Cherry and George Foster. 2012. Batch tuning strategies for statistical machine translation. In *Proceedings of the 2012 Conference of the North American Chapter of the ACL: Human Language Technologies*, pages 427–436, Montréal, Canada. ACL.

David Chiang, Yuval Marton, and Philip Resnik. 2008. Online large-margin training of syntactic and structural translation features. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 224–233. ACL.

David Chiang. 2007. Hierarchical phrase-based translation. *Computational Linguistics*, 33.

Michael Denkowski and Alon Lavie. 2011. Meteor 1.3: Automatic Metric for Reliable Optimization and Evaluation of Machine Translation Systems. In *Proceedings of the Sixth Workshop on Statistical Machine Translation*, Edinburgh, Scotland, July. ACL.

Jacob Devlin and Spyros Matsoukas. 2012. Trait-based hypothesis selection for machine translation. In *Proceedings of the 2012 Conference of the North American Chapter of the ACL: Human Language Technologies*, pages 528–532. ACL.

Kevin Duh, Katsuhito Sudoh, Xianchao Wu, Hajime Tsukada, and Masaaki Nagata. 2012. Learning to translate with multiple objectives. In *Proceedings of the 50th Annual Meeting of the ACL*, Jeju Island, Korea. ACL.

Chris Dyer, Hendra Setiawan, Yuval Marton, and Philip Resnik. 2009. The university of maryland statistical machine translation system for the fourth workshop on machine translation. In *Proc. of the Fourth Workshop on Machine Translation*.

Kevin Gimpel and Noah A. Smith. 2012. Structured ramp loss minimization for machine translation. In *Proceedings of the 2012 Conference of the North American Chapter of the ACL: Human Language Technologies*, pages 221–231, Montréal, Canada. ACL.

Keith Hall, Ryan T. McDonald, Jason Katz-Brown, and Michael Ringgaard. 2011. Training dependency parsers by jointly optimizing multiple objectives. In *Proceedings of the Empirical Methods in Natural Language Processing*, pages 1489–1499.

Xiaodong He and Li Deng. 2012. Maximum expected bleu training of phrase and lexicon translation models. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 292–301, Jeju Island, Korea. ACL.

Yifan He and Andy Way. 2009. Improving the objective function in minimum error rate training. In *MT Summit*.

Mark Hopkins and Jonathan May. 2011. Tuning as ranking. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, Edinburgh, Scotland. ACL.

Hideki Isozaki, Tsutomu Hirao, Kevin Duh, Katsuhito Sudoh, and Hajime Tsukada. 2010. Automatic evaluation of translation quality for distant language pairs. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, pages 944–952, Cambridge, MA. ACL.

Thorsten Joachims. 2006. Training linear svms in linear time. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 217–226.

Alon Lavie and Michael J. Denkowski. 2009. The meteor metric for automatic evaluation of machine translation. *Machine Translation*, 23(2-3):105–115.

Chang Liu, Daniel Dahlmeier, and Hwee Tou Ng. 2011. Better evaluation metrics lead to better machine translation. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*.

R. T. Marler and J. S. Arora. 2004. Survey of multi-objective optimization methods for engineering. *Structural and Multidisciplinary Optimization*, 26(6):369–395, April.

Arne Mauser, Saša Hasan, and Hermann Ney. 2008. Automatic evaluation measures for statistical machine translation system optimization. In *International Conference on Language Resources and Evaluation*, Marrakech, Morocco.

Franz Josef Och. 2003. Minimum error rate training in statistical machine translation. In *Proceedings of the 41st Annual Meeting of the ACL*, pages 160–167. ACL.

Sebastian Padó, Daniel Cer, Michel Galley, Dan Jurafsky, and Christopher D. Manning. 2009. Measuring machine translation quality as semantic equivalence: A metric based on entailment features. *Machine Translation*, 23(2-3):181–193.

Kishore Papineni, Salim Roukos, Todd Ward, and Wie-Jing Zhu. 2002. BLEU: a method for automatic evaluation of machine translation. In *Proceedings of Association of Computational Linguistics*, pages 311–318. ACL.

Majid Razmara, George Foster, Baskaran Sankaran, and Anoop Sarkar. 2012. Mixing multiple translation models in statistical machine translation. In *Proceedings of the 50th Annual Meeting of the ACL*, Jeju, Republic of Korea. ACL.

Baskaran Sankaran, Majid Razmara, and Anoop Sarkar. 2012. *Kriya* an end-to-end hierarchical phrase-based mt system. *The Prague Bulletin of Mathematical Linguistics (PBML)*, 97(97):83–98.

Christophe Servan and Holger Schwenk. 2011. Optimising multiple metrics with mert. *Prague Bull. Math. Linguistics*, 96:109–118.

Patrick Simianer, Stefan Riezler, and Chris Dyer. 2012. Joint feature selection in distributed stochastic learning for large-scale discriminative training in smt. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 11–21, Jeju Island, Korea. ACL.

Matthew Snover, Bonnie Dorr, Richard Schwartz, Linnea Micciulla, and John Makhoul. 2006. A study of translation edit rate with targeted human annotation. In *Proceedings of Association for Machine Translation in the Americas*, pages 223–231.

Valentin I. Spitkovsky, Hiyan Alshawi, and Daniel Jurafsky. 2011. Lateen EM: Unsupervised training with multiple objectives, applied to dependency grammar induction. In *Proceedings of the Empirical Methods in Natural Language Processing*, pages 1269–1280. Association of Computational Linguistics.

Taro Watanabe, Jun Suzuki, Hajime Tsukada, and Hideki Isozaki. 2007. Online large-margin training for statistical machine translation. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 764–773. ACL.

Taro Watanabe. 2012. Optimized online rank learning for machine translation. In *Proceedings of the 2012 Conference of the North American Chapter of the ACL: Human Language Technologies*, pages 253–262, Montréal, Canada, June. ACL.

Omar F. Zaidan. 2009. Z-MERT: A fully configurable open source tool for minimum error rate training of machine translation systems. *The Prague Bulletin of Mathematical Linguistics*, 91:79–88.

Daqi Zheng, Yifan He, Yang Liu, and Qun Liu. 2012. Maximum rank correlation training for statistical machine translation. In *MT Summit XIII*.