Improving the Objective Function in Minimum Error Rate Training

Yifan He and **Andy Way**
Centre for Next Generation Localisation
School of Computing
Dublin City University
Dublin 9, Ireland
{yhe,away}@computing.dcu.ie

Abstract

In Minimum Error Rate Training (MERT), the parameters of an SMT system are tuned on a certain evaluation metric to improve translation quality. In this paper, we present empirical results in which parameters tuned on one metric (e.g. BLEU) may not lead to optimal scores on the same metric. The score can be improved significantly by tuning on an entirely different metric (e.g. METEOR, by 0.82 BLEU points or 3.38% relative improvement on WMT08 English–French dataset).

We analyse the impact of choice of objective function in MERT and further propose three combination strategies of different metrics to reduce the bias of a single metric, and obtain parameters that receive better scores (0.99 BLEU points or 4.08% relative improvement) on evaluation metrics than those tuned on the standalone metric itself.

1 Introduction

Minimum Error Rate Training (MERT) (Och, 2003) tunes parameters in log-linear models by searching for the best parameter settings on the N-best output which minimizes translation errors according to automatic evaluation metrics such as BLEU (Papineni et al., 2002), METEOR (Banerjee and Lavie, 2005) or TER (Snover et al., 2006). In many shared translation tasks, it has been common practice to tune parameters with MERT against a mainstream evaluation metric to improve translation quality with respect to both automatic and human judgments. Most of the time the metric used in MERT is BLEU, but efforts have also been made to tune against other criteria (Dyer et al., 2009).

In this paper, we investigate the effect of the choice of metric, or the objective function in MERT. We find that in the single reference scenario (WMT08 English–French), tuning on BLEU leads to significantly inferior BLEU scores than when tuning on METEOR. We replicate similar results in another direction of the experiment with a slightly modified version of METEOR.

Based on this investigation, we propose three ways of combining different metrics into one error function to avoid the bias of single metrics: Linear Combination, and two methods of Constrained Search. These approaches receive scores on a par with or better than the best single metrics in the single reference scenario.

The rest of the paper is organised as follows. Section 2 reviews MERT, and Section 3 describes the characteristics of the different evaluation metrics we use. Sections 4 and 5 describe and analyse our experiments on single metrics as the objective functions in MERT. Sections 6 and 7 introduce our combination of different error functions and present experimental results. We conclude in Section 8, together with some avenues for further research.

2 Minimum Error Rate Training

Minimum Error Rate Training (MERT) is rooted in the log-linear models which have been applied successfully in SMT. In (Och and Ney, 2002) and (Koehn and Hoang, 2007), log-linear models are used to incorporate various information sources (features h_i) into the translation model, as in (1):

$$p(\mathbf{e}|\mathbf{f}) = \frac{1}{Z} exp \sum_{i=1}^{n} \lambda_i h_i(\mathbf{e}, \mathbf{f}) \quad (1)$$

MERT tunes the weights λ_i to minimize the errors on the error surface of the N-best list of the development set, as in (2):

$$\underset{\lambda}{\operatorname{argmin}} \; Err(e^*(\lambda); \mathbf{ref}) \quad (2)$$

In practice, the function Err is estimated by errors on a specific automatic evaluation metric m (most often BLEU). Then MERT is actually optimising on (3):

$$\underset{\lambda}{\operatorname{argmin}} \; err_m(e^*(\lambda); \mathbf{ref}) \quad (3)$$

Most current research has focused on the algorithm of MERT itself. For example, (Macherey et al., 2008) use word lattices instead of an N-best list to estimate the search space, and (Moore and Quirk, 2008) use random restarts to avoid local optima.

In this research, however, we try to improve the error surface/objective function on which MERT optimises. We will show that when the error surface cannot estimate the actual number of translation errors correctly; for example, when using BLEU with only a single reference as the objective function, the results can be improved by using an entirely different evaluation metric or a combination of different metrics during MERT.

3 Automatic MT Evaluation Metrics

Automatic evaluation metrics enable researchers to validate and optimize translation methods quickly. Simple *n*-gram-based metrics such as BLEU (Papineni et al., 2002) are fundamental to the development and tuning of MT systems and are widely applied in MERT.

However, it is well known that BLEU has many limitations (Callison-Burch et al., 2006). Many approaches have been proposed to overcome these insufficiencies, including METEOR (Banerjee and Lavie, 2005) and TER (Snover et al., 2006), both of which try to improve on the matching strategy used in BLEU.

There are other types of MT metrics that exploit deeper features such as paraphrases (Zhou et al., 2006), or syntax (Liu and Gildea, 2005; Owczarzak et al., 2007). There are also metrics that try to exploit machine learning techniques (Albrecht and Hwa, 2007; Ye et al., 2007; He and Way, 2009).

In this paper, we investigate the impact of using three representative metrics and their combinations as error functions in MERT. We have not tested with metrics that exploit deeper linguistic information because their computation is typically quite slow and thus less appropriate for MERT tuning.

3.1 BLEU

BLEU is the most popular evaluation metric in MT development. Although it suffers from several shortcomings, such as low correlation with human judgment on the sentence level, preference to statistical systems (Callison-Burch et al., 2006) and inconsistency in related evaluation scenarios (Chiang et al., 2008), it is still the automatic evaluation metric used in many translation campaigns and is often used as the error function in MERT.

BLEU performs *n*-gram matching between the output and the reference and the score is *n*-gram precision with a brevity penalty, as in (4):

$$\text{BLEU}(n) = \prod_{i=1}^{n} PREC_i^{\frac{1}{n}} \cdot bp \quad (4)$$

where n is the order of n-gram, $PREC_i$ is the i-gram precision and bp is the brevity penalty, as in (5):

$$bp = exp(max(\frac{len(Ref)}{len(Out)} - 1, 0)) \quad (5)$$

where $len(Ref)$ is the length of the reference and $len(Out)$ the length of the output. The *n*-gram matching scheme in BLEU makes it very sensitive to small changes in the output, especially in the single reference scenario. It has been shown in evaluation tasks (Callison-Burch et al., 2008) that BLEU has a lower correlation with human judgment than newer metrics like METEOR and TER.

3.2 METEOR

METEOR tries to solve the problems of BLEU by performing multi-stage unigram matching and adding recall as a consideration. With the use of unigram matching, METEOR is less sensitive to variations in word order, and with multi-stage matching, METEOR can consider stemming and WordNet semantic information. The METEOR score is calculated as in (6):

$$\text{METEOR} = \frac{PR}{\alpha P + (1-\alpha)R} \cdot (1 - cp) \quad (6)$$

where P is the unigram precision, R is the unigram recall and cp is the chunk penalty, calculated as in (7):

$$cp = \gamma \cdot (\frac{\#chunks}{\#matches})^{\beta} \quad (7)$$

METEOR set the different parameters α, β and γ for different target languages. This causes some of the inconsistencies in MERT that we demonstrate in Section 5.1.

3.3 TER

TER (Snover et al., 2006) is an Edit Distance-style evaluation metric. It calculates how many insertions, deletions, modifications and sequence shifts are needed to make the output and reference token sequences identical. The only difference between TER and classical Edit Distance (Levenshtein, 1966) is the sequence shift operation, which allows phrasal shifts in the output to be captured. TER is calculated as in (8):

$$\text{TER} = \frac{\#INS + \#DEL + \#MOD + \#SHIFT}{len(Ref)} \quad (8)$$

There is no explicit sentence-length penalty in TER, so the calculation of TER is based solely on counting edits/errors. As a result, we show in our experiments that TER prefers shorter sentences in MERT.

4 Experiments with Single Metrics

We first experiment with single metrics as the objective function in MERT. We treat single and multiple reference as two separate scenarios in what follows.

Among all the metrics, BLEU is more often used as the actual error function in MERT. The reasons for this might be due to BLEU's simplicity in computation and its status as the *de facto* automatic evaluation criterion in shared translation campaigns. However, our experiments show this choice to be questionable in the single reference scenario, such as WMT.

We tune on four single metrics: BLEU, METEOR, METEOR-SCP (Section 4.2) and TER, and evaluate the results on our testset with BLEU, METEOR and TER. We also report length ratio, which is the ratio between the output and the reference.

We also report the length ratio LEN of outputs, where n is the number of references, as in (9):

$$\text{LEN} = \frac{len(Output)}{\sum_i^n len(Ref_i)} \cdot n \quad (9)$$

As some metrics are biased to longer/shorter outputs, the length ratio helps us see whether a change in score is a real improvement, or rather a bias.

4.1 Experimental Settings

We conduct English–French and French–English single reference experiments on WMT 2008 data (WMT08). We use the top-1000 sentences in the original development set as our development set and the remaining 1000 as the test set. We train the translation model and a 4-gram language model on 1,288,074 sentence-pairs from Europarl (Koehn, 2005). The multiple reference experiments run on NIST 2006 (MT06) data. The translation model and a 3-gram language model is trained on data provided by LDC.

The single reference results are given in Tables 1 and 2, with the multiple reference results shown in Table 3. Scores in bold are statistically (1,000 bootstraps, 300 sentences each bootstrap) better than the others listed. We report results on both dev and test sets for the WMT08 dataset. All other results are test set results due to limited space.

In all experiments, we tune our parameters using a modified version of ZMERT (Zaidan, 2009) on the 100-best list generated by the phrase-based decoder Moses (Koehn et al., 2007) with 20 start points, the default setting in ZMERT to avoid local optima.

We use NIST BLEU 11b,[1] METEOR 0.7[2] and TER 0.725[3] as implementations of evaluation metrics. We do not use WordNet synonyms in METEOR.

[1] http://www.itl.nist.gov/iad/mig//tests/mt/2008/scoring.html

[2] http://www.cs.cmu.edu/˜alavie/METEOR/

[3] http://www.cs.umd.edu/˜snover/tercom/

Table 1: Experimental Results WMT08 English–French. MET: METEOR. LEN: Length Ratio. Rows are tuning criterion, columns are evaluation scores on test set.

	Dev set				Test set			
	BLEU	MET	TER	LEN	BLEU	MET	TER	LEN
BLEU (B)	**0.3022**	0.2169	0.5744	104%	0.2429	0.1763	0.6198	99%
MET (M)	**0.3061**	**0.2214**	0.5586	105%	**0.2511**	**0.1829**	0.6032	96%
TER (T)	0.2902	0.2163	**0.5392**	96%	0.2392	0.1782	**0.5924**	89%

Table 2: Experimental Results WMT08 French–English. MET: METEOR. MSCP: METEOR-SCP. LEN: Length Ratio.

	Dev set				Test set			
	BLEU	MET	TER	LEN	BLEU	MET	TER	LEN
BLEU (B)	**0.3070**	0.5449	**0.5404**	100%	**0.3276**	0.5552	**0.5252**	100%
MET (M)	0.2938	**0.5558**	0.5697	108%	0.3142	**0.5638**	0.5548	109%
TER (T)	0.2735	0.5258	**0.5396**	92%	0.2946	0.5373	**0.5255**	93%
MSCP	**0.3113**	**0.5540**	**0.5382**	102%	**0.3294**	**0.5631**	**0.5255**	103%

Just before we submitted this paper, METEOR released version 0.8 with the addition of a length penalty, which can serve a similar purpose to METEOR with a static chunk penalty (METEOR-SCP, in our terms) in MERT. We plan to use this feature in our future experiments.

4.2 METEOR with Static Chunk Penalty

One of the reasons why tuning on METEOR does not lead to high BLEU scores on French–English WMT data is that tuning on METEOR results in verbose translations. The output sentences are around 9% longer than the references. However, in the reverse direction, this does not happen and the length of the output is in the normal range.

We assume that this is caused by the different chunk penalties that METEOR assigns to different languages. For French, γ (cf. (7) above) is 1.0, but for English γ is 0.28. To fix this, we set a *static* $\gamma = 1.0$ for all target languages. The method is labeled as METEOR with static chunk penalty, METEOR-SCP.

The results in Table 2 show that tuning on METEOR-SCP leads to 1.52 points (4.84%) better BLEU scores than tuning on METEOR, and 0.18 points (0.55%) higher BLEU score than tuning on BLEU. It shows that the chunk penalty fixes METEOR's bias towards longer outputs to some extent, and at the same time preserves METEOR's better predictive power of translation quality than BLEU.

Table 3: Experimental Results MT06 Chinese-English. MET: METEOR. MSCP: METEOR-SCP. LEN: Length Ratio.

	BLEU	METEOR	TER	LEN
BLEU (B)	**0.2007**	0.4547	0.6962	102%
MET (M)	0.1766	**0.4640**	0.8197	121%
TER (T)	0.1847	0.4103	**0.6354**	77%
MSCP	**0.1931**	**0.4648**	0.7611	113%

5 Analysis and Discussion on Single Metrics Experiments

5.1 Single Reference Scenario

We can interpret the results in three respects. Firstly, the quality of the tuned parameters is dependent on the quality of the evaluation metric. In the English–French direction, tuning on METEOR can produce significantly better BLEU scores than tuning on BLEU itself. In the reverse direction, METEOR-SCP (Section 4.2) is statistically inferior to none of the three standard metrics used in our experiments. Our interpretation is that METEOR has better predictive power of translation quality than BLEU, so it serves as a more consistent error function during training.

Secondly, metrics are often biased to informative or precise sentences, which prevent some metrics from making correct judgments on translation quality during tuning. TER, for instance, prefers short outputs. In the following example, HYP2 will receive a lower (i.e.better) TER score than HYP1, because it uses 2 insertions to avoid shifts and deletions.

REF: *The house is small*

HYP1: *That is a small house* (TER: 0.75)

HYP2: *The house* (TER: 0.5)

Such an example is less likely to harm in MT evaluation unless a system is developed specifically to game the metric. However, if we make use of such knowledge in tuning, the system will be *tuned* to take advantage of this preference, and will tend to output overly succinct sentences.

In both experiments, tuning on TER leads to results that obtain the best TER, but the worst BLEU and METEOR scores. METEOR suffers from a similar problem in the French–English direction, because TER favours precise sentences while METEOR favours informative sentences.

The length ratios of outputs tuned with different metrics are listed in the tables beside the scores from the specific evaluation metrics. In our French–English experiments, TER generates outputs that are 7% shorter than the reference, and METEOR generates 9% longer translations. For example, METEOR- and TER-oriented tuning give the following output for the same input sentence:

REF: *it is important that our products are safe , but we should not go over the top with extreme actions .* (21 tokens)

METEOR-TUNED: *it is important that our products are safe , but we must be careful not to go beyond certain limits as a combination of extreme actions .* (27 tokens)

TER-TUNED: *it is important that our products are safe , but we must not exceed certain limits extreme through actions .* (20 tokens)

We introduced METEOR-SCP in Section 4.2 to address this problem.

Thirdly, it is questionable whether we should continue to use BLEU as the metric for tuning in single reference scenarios such as WMT08, even when our aim is simply to improve on the BLEU score itself, regardless of any actual improvements in translation quality. In previous shared tasks of WMT, there are submissions that use other metrics for tuning (e.g. (Dyer et al., 2009)) in order to achieve higher correlation with human judgment. In our experiments, however, tuning on METEOR or METEOR-SCP can be better than tuning on BLEU even if our aim is to obtain a higher BLEU score.

Finally, it is interesting to see that the results on both the dev and test sets follow the same pattern in our experiments. One might expect that tuning on BLEU should lead to the better BLEU scores on the tuning set, as there is no danger of overfitting. Our results, in which METEOR or its SCP variant outperforms BLEU on the dev set, may suggest that the problem of using single-reference BLEU in MERT is not overfitting, but the incorrect error surface that hinders MERT from finding the optimal parameters.

5.2 Multiple Reference Scenario

In this scenario, optimising on each metric will produce the best scores on that metric. We suspect that multiple references improve the estimation power of the evaluation metrics and generate more stable results.

Though METEOR-SCP cannot produce higher BLEU scores than tuning on BLEU as in the single reference scenario, it still improves upon the original version of METEOR in both BLEU and TER scores, and the length ratio is more acceptable. It even receives a 0.08 points better (original) METEOR score. These results again show how the default value of γ in METEOR causes a bias to verbose outputs during tuning.

Besides, multiple references amplify the bias of METEOR and TER towards longer/shorter sentences. METEOR with the original chunk penalty leads to outputs that can be 21% longer than the references, while TER can cause outputs to be up to 23% shorter. In such cases, these biases hinder these two metrics—believed to have better predictive power than BLEU—from materializing their advantage over BLEU in MERT.

Multiple references also give BLEU better predictive power. In (Papineni et al., 2002), the single means of capturing variation in translation is to use multiple references. This might explain why BLEU is the more consistent metric among the four baseline metrics in our multiple reference experiments.

6 Combining Evaluation Metrics in MERT

We propose to combine various metrics to overcome the bias and inconsistency of single metrics. We introduce and evaluate on three types of combination: linear combination, metric constraint search and length constraint search. The experimental settings are the same as for the single metric experiments.

6.1 Linear Combination

Linear combination uses the sum of a set of evaluation metrics as the error function, as in (10):

$$\operatorname*{argmin}_{\lambda}\, err_{m(1)+\cdots+m(n)}(e^{*}(\lambda); \mathbf{ref}) \qquad (10)$$

where $m(1)$ to $m(n)$ are different automatic evaluation metrics. Linear combination is reported to be used in MERT (Dyer et al., 2009) and different weights are set for each metric $m(i)$. However, from what can be discerned from the paper, the choice of weights would appear to be arbitrary.

6.2 Metric Constrained Search

Instead of optimising on the linear combination of metrics, we can continue to optimise on a single metric, and reduce the arbitrariness of that metric using constraints, as in (11):

$$\operatorname*{argmin}_{\lambda}\, err_{m(0)}(e^{*}(\lambda); \mathbf{ref}) \qquad (11)$$

$$\begin{aligned} s.t. \quad err_{m(1)}(e^{*}(\lambda); \mathbf{ref}) &\leqslant err^{curr}_{m(1)}(e^{*}(\lambda); \mathbf{ref}) \\ &\vdots \\ err_{m(n)}(e^{*}(\lambda); \mathbf{ref}) &\leqslant err^{curr}_{m(n)}(e^{*}(\lambda); \mathbf{ref}) \end{aligned}$$

where $m(0)$ is the metric to tune on, and $m(1)$ to $m(n)$ are the constraint metrics.

In the constrained search, MERT keeps optimising on the metric $m(0)$. However, when choosing the λ with the minimum error on $m(0)$, the algorithm is constrained by constraint metrics $m(1), \cdots, m(n)$, so that the number of errors measured by these metrics should not increase.

6.3 Results

In the tables below, we report the results tuned on different combination strategies. Datasets and other settings are the same as for the single metric experiments. For every test setting, we again report the BLEU, METEOR and TER scores as well as the length ratio (LEN). Bold scores are significantly better than others reported. In the tables $m(0) + m(1)$ represents a linear combination of $m(0)$ and $m(1)$ (Section 6.1), while $m(0)/m(1)$ represents tuning on $m(0)$ with $m(1)$ as a constraint (Section 6.2). $Single^{*}$ are the best metric scores tuned on any standalone metric.

Table 4: Experimental Results WMT08 English–French. MET: METEOR. LEN: Length Ratio. Rows are tuning criterion, columns are evaluation scores on test set.

	BLEU	MET	TER	LEN
$Single^{*}$	**0.2511**	**0.1829**	**0.5924**	-
B+M	**0.2528**	**0.1821**	0.6040	97%
B+T	0.2475	**0.1821**	**0.5913**	92%
B+M+T	**0.2492**	**0.1814**	0.5991	94%
B/M	0.2360	0.1704	0.6413	100%
B/T	**0.2508**	**0.1811**	0.6178	99%

Table 5: Experimental Results WMT08 French–English. MET: METEOR. LEN: Length Ratio.

	BLEU	MET	TER	LEN
$Single^{*}$	0.3294	**0.5638**	0.5252	-
B+M	0.3324	0.5577	**0.5157**	100%
B+T	0.3201	0.5500	**0.5122**	96%
B+M+T	0.3098	0.5452	0.5270	97%
B/M	0.3016	0.5417	0.5331	96%
B/T	**0.3386**	**0.5609**	**0.5142**	101%

Table 6: Experimental Results MT06 Chinese–English. MET: METEOR. LEN: Length Ratio.

	BLEU	METEOR	TER	LEN
$Single^{*}$	**0.2071**	**0.4648**	**0.6354**	-
B+M	**0.2013**	**0.4644**	0.7357	109%
B+T	**0.2051**	0.4342	0.6521	90%
B+M+T	**0.2034**	0.4477	0.6774	96%
B/M	**0.2015**	0.4607	0.7301	108%
B/T	0.1943	0.4598	0.7179	106%

7 Analysis and Discussion of Combined Metrics Experiments

As stated in Section 5, tuning on a single metric, especially one with limited predictive power of translation quality (e.g. single reference BLEU), does not lead to optimal scores on that metric. In this section we analyse the effect of using a combination of metrics as the error function in MERT.

7.1 Single Reference Scenario

In our experiments, using simple linear combination can already improve automatic evaluation scores. In both directions for our WMT08 experiments, the linear combination of BLEU and METEOR achieves higher BLEU scores than tuning on BLEU alone. In the English–French direction, the BLEU score is 0.17 (0.6%) points higher and in the opposite direction it is 0.3 (0.9%) points higher. Furthermore, the differences are often insignificant when the scores of the combined methods are worse.

Moreover, TER-constrained BLEU works best among all the combination strategies on the single reference experiments. In the English–French direction, it receives BLEU and METEOR scores that are statistically on a par with the best single metrics. In the French–English direction, it yields the highest BLEU score (by 0.92 points, or 2.79%) which is significantly better than any other configuration, and the METEOR and TER scores are still on a par with the best scores.

From another perspective, the length ratio of tuning on combined metrics is much closer to 1, which indicates that our proposed methods serve the purpose of avoiding the length bias of different metrics.

7.2 Multiple Reference Scenario

In the multi-reference scenario, however, it is much harder to obtain translations that are on a par or better than the single best translations on every metric, because the single best result is often achieved for translations that differ considerably with respect to sentence length compared to the reference, e.g. the single best TER score is obtained on a 23% shorter translation than the reference.

The combinations do not outperform any single best scores. However, the linear combination of BLEU and METEOR yields BLEU and METEOR scores on a par with the single best.

The length ratio is better than tuning on a single metric, but is still far from the "rational" range. The output from the combination of BLEU and METEOR, for example, is just 9% longer than the reference.

8 Conclusions and Future Work

In this paper, we explore the effect of tuning on different objective functions in Minimum Error Rate Training, and designed two combinations of metrics to improve the error function in MERT.

In the single metric experiments, we show that tuning on BLEU yields worse BLEU scores than tuning on METEOR or its variants when we have only one reference. Of course, BLEU is not designed for the single reference scenario and in such a case, it may be wiser to rely on metrics with a better ability to capture variances, such as METEOR.

In the combination of metrics experiments, the linear combination of BLEU and METEOR/TER and TER-constrained BLEU all yield more consistent results, which are statistically on a par or better than the best scores tuned on single metrics.

Our methods do not work as well in the multiple reference scenarios, because multiple references enable better prediction from BLEU, and worsen the bias of the different metrics.

Our linear combination method has some limitations: there is no reason why the scores of several different evaluation metrics could not be added, and we do not know how to set the proper weights to the metrics when we add them. All these questions are open for future research. In addition, it would be interesting to see whether we can find ways to avoid bias towards certain types of sentences when designing new metrics for MT evaluation.

Acknowledgments

We are grateful to Science Foundation Ireland (http://www.sfi.ie) grant number 07/CE/I1142 for sponsoring this research.

References

Joshua Albrecht and Rebecca Hwa. 2007. Regression for sentence-level MT evaluation with pseudo references. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 296–303, Prague, Czech Republic.

Satanjeev Banerjee and Alon Lavie. 2005. METEOR: An automatic metric for MT evaluation with improved correlation with human judgments. In *Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evalu-*

ation Measures for Machine Translation and/or Summarization, pages 65–72, Ann Arbor, MI.

Chris Callison-Burch, Cameron Fordyce, Philipp Koehn, Christof Monz, and Josh Schroeder. 2008. Further meta-evaluation of machine translation. In *Proceedings of the Third Workshop on Statistical Machine Translation*, pages 70–106, Columbus, OH.

Chris Callison-Burch, Miles Osborne, and Philipp Koehn. 2006. Re-evaluating the role of Bleu in machine translation research. In *EACL 2006, 11th Conference of the European Chapter of the Association for Computational Linguistics, Proceedings*, pages 249–256, Trento, Italy.

David Chiang, Steve DeNeefe, Yee Seng Chan, and Hwee Tou Ng. 2008. Decomposability of translation metrics for improved evaluation and efficient algorithms. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*, pages 610–619, Honolulu, HI.

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 *Proceedings of the Fourth Workshop on Statistical Machine Translation*, pages 145–149, Athens, Greece.

Yifan He and Andy Way. 2009. Learning labelled dependencies in machine translation evaluation. In *Proceedings of EAMT-09, the 13th Annual Meeting of the European Association for Machine Translation*, pages 44–51, Barcelona, Spain.

Philipp Koehn. 2005. Europarl: A Parallel Corpus for Statistical Machine Translation. In *Proceedings of MT Summit X*, pages 79–86, Phuket, Thailand.

Philipp Koehn and Hieu Hoang. 2007. Factored translation models. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 868–876, Prague, Czech Republic.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondrej Bojar, Alexandra Constantin, and Evan Herbst. 2007. Moses: Open source toolkit for statistical machine translation. In *45th Annual Meeting of the Association for Computational Linguistics, Companion Volume: Proceedings of the Demo and Poster Sessions*, pages 177–180, Prague, Czech Republic.

Vladimir Iosifovich Levenshtein. 1966. Binary codes capable of correcting deletions, insertions, and reversals. *Soviet Physics Doklady*, 10(8):707–710.

Ding Liu and Daniel Gildea. 2005. Syntactic features for evaluation of machine translation. In *Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*, pages 25–32, Ann Arbor, MI.

Wolfgang Macherey, Franz Och, Ignacio Thayer, and Jakob Uszkoreit. 2008. Lattice-based minimum error rate training for statistical machine translation. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*, pages 725–734, Honolulu, HI.

Robert C. Moore and Chris Quirk. 2008. Random restarts in minimum error rate training for statistical machine translation. In *Proceedings of the 22nd International Conference on Computational Linguistics (Coling 2008)*, pages 585–592, Manchester, UK.

Franz Josef Och. 2003. Minimum error rate training in statistical machine translation. In *41st Annual Meeting on Association for Computational Linguistics*, pages 160–167, Sapporo, Japan.

Franz Josef Och and Hermann Ney. 2002. Discriminative training and maximum entropy models for statistical machine translation. In *40th Annual Meeting of the Association for Computational Linguistics*, pages 295–302, Philadelphia, PA.

Karolina Owczarzak, Josef van Genabith, and Andy Way. 2007. Labelled dependencies in machine translation evaluation. In *Proceedings of the Second Workshop on Statistical Machine Translation*, pages 104–111, Prague, Czech Republic.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, PA.

Matthew Snover, Bonnie Dorr, Richard Schwartz, Linnea Micciulla, and John Makhoul. 2006. A study of translation edit rate with targeted human annotation. In *AMTA 2006, Proceedings of the 7th Conference of the Association for Machine Translation in the Americas*, pages 223–231, Cambridge, MA.

Yang Ye, Ming Zhou, and Chin-Yew Lin. 2007. Sentence level machine translation evaluation as a ranking. In *Proceedings of the Second Workshop on Statistical Machine Translation*, pages 240–247, Prague, Czech Republic.

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.

Liang Zhou, Chin-Yew Lin, Dragos Stefan Munteanu, and Eduard Hovy. 2006. Paraeval: Using paraphrases to evaluate summaries automatically. In *Proceedings of the Human Language Technology Conference of the NAACL, Main Conference*, pages 447–454, New York City, NY.