Expected Error Minimization with Ultraconservative Update for SMT

Lemao LIU[1], *Tiejun ZHAO*[1], *Taro WATANABE*[2], *Hailong CAO*[1], *Conghui ZHU*[1]
(1) School of Computer Science and Technology
Harbin Institute of Technology, Harbin, China
(2) National Institute of Information and Communication Technology
3-5 Hikari-dai, Seika-cho, Soraku-gun, Kyoto, Japan
{lmliu,tjzhao,hailong,chzhu}@mtlab.hit.edu.cn, taro.watanabe@nict.go.jp

ABSTRACT
Minimum error rate training is a popular method for parameter tuning in statistical machine translation (SMT). However, the optimization objective function may change drastically at each optimization step, which may induce MERT instability. We propose an alternative tuning method based on an ultraconservative update, in which the combination of an expected task loss and the distance from the parameters in the previous round are minimized with a variant of gradient descent. Experiments on test datasets of both Chinese-to-English and Spanish-to-English translation show that our method can achieve improvements over MERT under the Moses system.

KEYWORDS: statistical machine translation; tuning; minimum error rate training; ultraconservative update; expected BLEU.

Proceedings of COLING 2012: Posters, pages 723–732,
COLING 2012, Mumbai, December 2012.

1 Introduction

Minimum error rate training (Och, 2003), MERT, is an important component of statistical machine translation (SMT), and it has been the most popular method for tuning parameters for SMT systems. One of its major contributions is the use of an evaluation metric, such as BLEU (Papineni et al., 2002), as a direct loss function during its optimization procedure by interchanging decoding and optimization steps in each round.

While MERT is successful in practice, it is known to be unstable (Clark et al., 2011). At the optimization step in each round, MERT tries to repeatedly optimize a loss function defined by the k-best candidate lists. Since new k-best lists are generated and merged with the previously generated lists at each round, the optimization objective function may change drastically between two adjacent rounds (Pauls et al., 2009), and the optimized weights of these two rounds may also be far from each other.

Motivated by the above observation, this paper investigates a new tuning approach under the k-best lists framework, instead of the lattices or hypergraphs framework as Macherey et al. (2008) and Kumar et al. (2009), to achieve a more stable loss function between optimization steps. We propose an expected loss-based ultraconservative update method, in which an expected loss is minimized using an ultraconservative update strategy (Crammer and Singer, 2003; Crammer et al., 2006). In the optimization step, we iteratively learn the weight which should not only minimize the error rates as in MERT but also not be far from the weight learned at the previous optimization step. Instead of using the L_2 in Euclidean space to describe the distances between the two weights as in the Margin Infused Relaxed Algorithm (MIRA), we define a new distance metric inspired by the max-posterior probability decoding strategy in translation.

Compared with MERT, in which an exact line search is difficult to implement, our method is easier since we employ a gradient-based algorithm, which is simple but proved to be successful in other tasks such as tagging or parsing. Further, experiments on Chinese-to-English and Spanish-to-English show that our method outperforms MERT.

2 MERT Revisited

MERT is the most popular method to tune parameters for SMT systems. The main idea behind it is that it iteratively optimizes the weight such that, after re-ranking a k-best list of a given development set with this weight, the error of the resulting 1-best list is minimal.

The whole tuning algorithm with MERT is described in Algorithm 1. It requires a development set $\{\langle f_i; \mathbf{r}_i \rangle\}_{i=1}^n$ with f_i as the source sentence and $\mathbf{r}_i$ as its reference, initial weight W_{init} and the maximal iterations K. It initializes some parameters in line 1: iteration index k; the current weight W_k; the accumulated k-best list $\mathbf{c}_i$. For each optimization step $k \leq K$, it repeatedly performs decoding and training during the loop from line 2 to line 9: for each sentence f_i, it decodes to get $\mathbf{tc}_i$ and updates $\mathbf{c}_i$; it minimizes the error rates to obtain W_{k+1}. At the end of the algorithm, it returns W_K.

The definition of **Loss** in line 7 of Algorithm 1 is formalized as follows:

$$\mathbf{Loss}_{error}\Big(\big\{ \mathbf{r_i}; \hat{e}(f_i; W) \big\}_{i=1}^n \Big), \tag{1}$$

Algorithm 1 TUNING WITH MERT

```
Input: $\{\langle f_i; \mathbf{r}_i \rangle\}_{i=1}^{n}; W_{init}; K$
Output: $W$
$k = 1; W_k = W_{init}; \{\mathbf{c}_i = \emptyset\}_{i=1}^{n}$ //initialization, $\mathbf{c}_i$ the accumulated k-best list for $f_i$
while $k \leq K$ do
    for all sentence $f_i$ such that $1 \leq i \leq n$ do
        Decode $f_i$ with $W_k$ to get $\mathbf{tc}_i$; // $\mathbf{tc}_i$ translation candidates of k-best decoding
        $\mathbf{c}_i = \mathbf{c}_i \bigcup \mathbf{tc}_i$;
    end for
    Set $W_{k+1}$ as the weight according to a Loss of error rates defined on $\mathbf{tc}_i$ and $W$;
    $k$++;
end while
$W = W_K$;
```

with

$$\begin{aligned}\hat{e}(f;W) &= \text{argmax}_e \mathbf{P}(e|f;W) \\ &= \text{argmax}_e \frac{\exp\left\{W \cdot h(f,e)\right\}}{\sum_{e'} \exp\left\{W \cdot h(f,e')\right\}} = \text{argmax}_e \left\{W \cdot h(f,e)\right\},\end{aligned} \tag{2}$$

where $h(f,e)$ denotes the feature vector of f and its translation e. $\mathbf{Loss}_{error}$ is usually set as Corpus-BLEU (exactly speaking, minus BLEU). Eq. 2 describes the maximal posterior decoding strategy.

As mentioned in Section 1, since at each optimization step a new k-best list $\mathbf{tc}_i$ is generated and merged with $\mathbf{c}_i$, the optimization objective will change between two adjacent optimization steps. This can increase the instability of MERT. In the next section, we will investigate the strategy of ultraconservative update to address this issue.

3 Expected Loss Based Ultraconservative Update

Ultraconservative Update is an efficient way to consider the trade-off between the amount of progress made on each round and the amount of information retained from previous rounds. On one hand, the weight update should assure better performance to improve optimization. On the other hand, the new weight must stay as close as possible to the weight optimized on the last round, thus retaining the information learned on previous rounds.

3.1 Objective Function

Suppose W_k be the weight learnt from last optimization step, $\{\langle f_i; \mathbf{c}_i; \mathbf{r}_i \rangle\}_{i=1}^{n}$ a translation space obtained with W_k, where f_i is a source sentence, $\mathbf{c}_i$ is a set of translation candidates and $\mathbf{r}_i$ is a set of references for f_i. Now we want to optimize W_{k+1} using the idea of ultraconservative update to the objective of MERT, and we obtain the following objective function:

$$\mathbf{d}(W, W_k) + \lambda \mathbf{Loss}_{error}\Big(\left\{\mathbf{r_i}; \hat{e}(f_i; W)\right\}_{i=1}^{n}\Big), \tag{3}$$

where $\mathbf{d}(W, W_k)$ is a distance function of a pair of weights and it is used to penalize a weight far away from W_k. $\mathbf{Loss}_{error}$ is the objective function of MERT as defined in Eq. 1. $\lambda \geq 0$ is the regularization penalty. When $\lambda \to \infty$ Eq. 3 goes back to the objective function of MERT.

Because the first term **d** in Eq. 3 is not piecewise linear in respect to W, the exact line search routine in MERT does not hold anymore. Generally, it is not easy to directly minimize Eq. 3. Motivated by (Och, 2003; Smith and Eisner, 2006; Zens et al., 2007), we use the expected loss to substitute the direct loss in Eq. 3 and we obtain the objective function as follows:

$$\mathbf{d}(W, W_k) + \frac{\lambda}{n}\sum_{i=1}^{n}\sum_{e\in\mathbf{c}_i}\mathbf{Loss}_{error}(\mathbf{r}_i; e)\mathbf{P}_\alpha(e|f_i; W), \tag{4}$$

with

$$\mathbf{P}_\alpha(e|f_i; W) = \frac{\exp[\alpha W \cdot h(f_i, e)]}{\sum_{e'\in\mathbf{c}_i}\exp[\alpha W \cdot h(f_i, e')]},$$

where $\alpha > 0$ is a real number, each $h(f_i, e)$ is a feature vector, and **d** is a distance metric defined on a pair of weights. $\mathbf{Loss}_{error}(\mathbf{r}_i; e)$ in Eq. 4 is a sentence-wise direct loss, and in this paper we used a variant of sentence BLEU proposed by Chiang et al. (2008) which smoothes BLEU statistics with pseudo-document.

3.2 Distance Metric Based on Projection

Euclidean distance (L_2 norm) is usually employed as in MIRA (Watanabe et al., 2007; Chiang et al., 2008). In this section we will specifically investigate another metric for ultraconservative update in SMT.

In log-linear based translation models, since the decoding strategy is the maximal posterior probability, the translation results are the same for the weight W and its positive multiplication (see Eq. 2). Therefore, for a translation decoder, we wish that the distance of two weights satisfies the following property: the smaller the distance between them is, the more similar the translation results decoded with them are. However, L_2 norm does not satisfy this property. Inspired by this observation, we define the distance[1] between W and W' as follows:

$$\mathbf{d}(W, W') = \begin{cases} 0, \text{ either } W \text{ or } W' \text{ is } 0 & , \\ \frac{1}{2}\|\frac{W}{\|W\|} - \frac{W'}{\|W'\|}\|^2, \text{ otherwise} & , \end{cases} \tag{5}$$

For the sake of simplicity, if we constrain the feasible region to $\{W : \|W\| = 1\}$ and substitute the above **d** in Eq. 4, we derive the following optimization problem:

$$\begin{aligned} &\min_W \left\{ \frac{1}{2}\|W - W_k\|^2 + \frac{\lambda}{n}\sum_{i=1}^{n}\sum_{e\in\mathbf{c}_i}\mathbf{Loss}_{error}(\mathbf{r}_i; e)\mathbf{P}_\alpha(e|f_i; W) \right\} \\ s.t. & \\ &\|W\| = 1, \end{aligned} \tag{6}$$

where we assume $\|W_k\| = 1$, otherwise we can normalize it instead. Since Eq. 6 is defined on the expected loss and ultraconservative update, we call it the expected loss based ultraconservative update, or ELBUU.

[1] Strictly speaking, it is not the traditional distance metric because it violates the property of positive definiteness. For example, when one of W and W' is zero and the other is not, it does not hold that $\mathbf{d}(W, W') = 0$ induces $W = W'$. However, in this paper, our attention is focused on the non-zero weights.

3.3 Gradient Descent with Projection

We employ the gradient projection method (Horst and Tuy, 1996) to optimize Eq. 6. The gradient projection method contains two main operations, one of which is the gradient descent for the objective function and the other is the projection of the weight into the constraint area. The first operation is easy to implement. For the second one, taking the derivative of $\mathbf{P}_\alpha(e|f_i;W)$ with respect to W , the following equation holds:

$$\nabla_W \mathbf{P}_\alpha(e|f;W) = \alpha \mathbf{P}_\alpha(e|f;W)\Big(h(f,e) - E_{\mathbf{P}_\alpha(\cdot|f;W)}(h(f,\cdot)\Big), \tag{7}$$

with

$$E_{\mathbf{P}_\alpha(\cdot|f;W)}(h(f,\cdot)) = \sum_{e'} \mathbf{P}_\alpha(e'|f;W) * h(f,e'),$$

where $E_{\mathbf{P}_\alpha(\cdot|f;W)}$ can be interpreted as the expectation of feature function $h(f,\cdot)$ according to the distribution of $\mathbf{P}_\alpha(\cdot|f;W)$. Then, the derivative of the objective function in Eq. 6 is as follows:

$$\Delta = W - W_k + \frac{\lambda}{n}\sum_{i=1}^{n}\sum_{e} \mathbf{Loss}_{error}(\mathbf{r}_i, e)\nabla_W \mathbf{P}_\alpha(e|f;W). \tag{8}$$

Algorithm 2 gives the pseudo-code of the gradient projection method to optimize Eq. 6. In the Algorithm,$\eta > 0$ is the learning rate , $\epsilon > 0$ is the threshold, and other notations are the same as before. The loop (line 2-10) is the whole iteration step, which contains a gradient descent operation in line 3 and a projection operation[2] in line 4-8. At the end of this algorithm, it returns W_{k+1}.

Algorithm 2 Gradient Descent with Projection

```
Input: $W_k, \lambda, \epsilon, \alpha, \eta,$
Output: $W_{k+1}$
 1: $W_k^0 \neq W_k; W_k^1 = W_k; t = 1; \eta_1 = 1/\eta;$ // initialization
 2: while ($||W_k^t - W_k^{t-1}|| > \epsilon$) do
 3:     $W_k^{t+1} = W_k^t - \eta_t \Delta$ according to Eq. 8; // gradient operation
 4:     if $W_k^{t+1} \neq 0$ then
 5:         $W_k^{t+1} = W_k^{t+1}/||W_k^{t+1}||$; //projection operation
 6:     else
 7:         Reset $W_k^{t+1}$ $s.t.$ $||W_k^{t+1}|| = 1$;
 8:     end if
 9:     $t{+}{+}; \eta_t = 1/(\eta \cdot t)$;
10: end while
11: $W_{k+1} = W_k^t$;
```

[2] Actually, in our experiments, W does not arrive at the point 0 during the iteration steps.

Methods	NIST02(Dev)	NIST03	NIST04	NIST05	NIST06	NIST08
MERT	30.39	26.45	29.47	26.31	25.34	19.07
ELBUU	30.06	27.36^{++}	29.89	27.03^{+}	26.30^{++}	19.79^{+}

Table 1: Comparison of two tuning methods, MERT and ELBUU, on Chinese-to-English translation tasks. $+$ or $++$ means the ELBUU method is significantly better than MERT with confidence $p < 0.05$ or $p < 0.01$, respectively.

3.4 Tuning with ELBUU

Similar to tuning algorithm MERT, i.e. **Algorithm** 1, our tuning algorithm ELBUU repeatedly performs decoding and optimization. In detail, Our ELBUU can be obtained from **Algorithm** 1 as follows: by inserting the **Algorithm** 2 to substitute for line 7 in **Algorithm** 1, and modifying the returned weight as averaged weight[3] at the end of the algorithm, one can obtain the ELBUU tuning algorithm.

Our method ELBUU is similar to the MIRA in (Watanabe et al., 2007; Chiang et al., 2008), since both of them employ a strategy of ultraconservative update. However, there are also some differences between them. ELBUU optimizes the expected BLEU, a loss more approximate towards Corpus-BLEU compared with the generalized hinge loss, and it utilizes the projection distance metric instead of L_2 as with MIRA. Further, ELBUU is a MERT-like batch mode which ultraconservatively updates the weight with all training examples, but MIRA is an online one which updates with each example (Watanabe et al., 2007) or parts of examples (Chiang et al., 2008). The batch mode has some advantages over online mode: more accurate sentence-wise BLEU towards Corpus-BLEU (Watanabe, 2012) and more promising experimental performance (Cherry and Foster, 2012). Additionally, our method is similar to (Liu et al., 2012). However, the main difference is that ours is a global training method instead of a local training method.

4 Experiments and Results

4.1 Experimental Setting

We conduct our translation experiments on two language pairs: Chinese-to-English and Spanish-to-English. For the Chinese-to-English task, the training data is FBIS corpus consisting of about 240k sentence pairs; the development set is NIST02 evaluation data; the test set NIST05 is used as the development test set for tuning hyperparameter λ in Eq. 6; and the test datasets are NIST03, NIST04, NIST05, NIST06, and NIST08. For the Spanish-to-English task, all the datasets are from WMT2011: the training data is the first 200k sentence pairs of Europarl corpus; the development set is dev06; and the test datasets are test07, test08,test09, test10, test11.

We run GIZA++ (Och and Ney, 2000) on the training corpus in both directions (Koehn et al., 2003) to obtain the word alignment for each sentence pair. We train a 4-gram language model on the Xinhua portion of the English Gigaword corpus using the SRILM Toolkits (Stolcke, 2002) with modified Kneser-Ney smoothing (Chen and Goodman, 1998). In our experiments, the translation performances are measured by the case-insensitive BLEU4 metric (Papineni et al., 2002) and we use mteval-v13a.pl as the evaluation tool. The significance testing is performed by paired bootstrap re-sampling (Koehn, 2004).

[3] At the end of tuning, we average the weights as (Collins, 2002). The norm of the averaged weight may nolonger be equal to 1, but it is irrelevant for testing, see discussion in Section 3.2.

Methods	dev06(Dev)	test08	test09	test10	test11
MERT	28.85	19.68	21.36	23.35	23.65
ELBUU	28.67	20.23	21.72	23.90^{+}	24.18^{+}

Table 2: Comparison of two tuning methods, MERT and ELBUU, on Spanish-to-English translation tasks. + means the ELBUU method is significantly better than MERT with confidence $p < 0.05$.

Distance metrics	NIST02(Dev)	NIST03	NIST04	NIST05	NIST06	NIST08
L_2	29.95	27.09	29.65	26.79	25.98	19.54
Projection	30.06	27.36	29.89	27.03	26.30	19.79

Table 3: Comparison of two distance metrics L_2 and projection on Chinese-to-English translation tasks.

The translation system is a phrase-based translation model (Koehn et al., 2003) and we use the open source toolkit **MOSES** (Koehn et al., 2007) as its implementation. In the experiments, the default setting is used for MOSES. The baseline tuning method is the standard algorithm MERT and the k-best-list size is set as 100 for tuning. For ELBUU, we empirically set $\alpha = 3.0$ as (Och, 2003), $\eta = 1$, $\epsilon = 10^{-5}$, $K = 20$, and we do not tune them further. We tune λ on NIST05 with $\lambda = 1.0$ for the Chinese-to-English translation tasks and we do not tune it again for the Spanish-to-English translation tasks.

4.2 Results

Table 1 and Table 2 give the main results of ELBUU compared with the baseline MERT on Chinese-to-English and Spanish-to-English translation tasks, respectively. Overall, we can see that the proposed ELBUU achieves consistent improvements on both language pairs: ELBUU is better than MERT, although some of the comparisons are not significant. In detail, for Chinese-to-English tasks, ELBUU achieves improvements from 0.42 BLEU points on NIST04 to 0.96 BLEU points on NIST06; and for Spanish-to-English tasks, ELBUU also outperforms MERT with improvements up to 0.5 BLEU points on both the test10 and test11 test sets.

Table 3 shows the performance of the distance metric defined in section 3.3, and L_2 is used as its comparison[4]. We also tune it on NIST05 and set it to 0.1 for the case of L_2 distance. Although the comparison results are not significant, we can see that the performance of projection distance is slightly better than that of L_2 distance.

Figure 1 shows the learning curves during tuning for Chinese-to-English translation tasks. It shows that the performances over the test datasets do not decrease as iterations increase and the weights can achieve stable performances within 20 iterations.

To further testify to the advantage of the ultraconservative update, we fix the k-best-list results as those produced by MERT and compare ELBUU with MERT: when running ELBUU, we do not perform the decoding step to generate the k-best list $\mathbf{tc}_i$, and instead we set it as the k-best list

[4] The algorithm of ELBUU with L_2 as its distance is the same as ELBUU with projection distance after deleting the projection step in line 4-8 of Algorithm 2

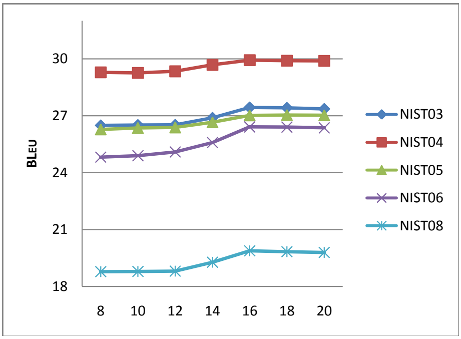

Figure 1: The learning curves for ELBUU as tuning algorithm on all the test sets of Chinese-to-English translation. The horizontal axis denotes the number of iterations during tuning, and the vertical one denotes the BLEU points.

Methods	NIST05	NIST06	NIST08
MERT	26.31	25.34	19.07
ELBUU	26.65	25.85	19.41

Table 4: The comparison of ELBUU and MERT with the same k-best-list results for optimization under the Chinese-to-English translation tasks.

exactly obtained by MERT tuning at the corresponding decoding step. Table 4 shows that the ELBUU is slightly better than MERT. This fact also directly indicates the advantages of ELBUU over MERT.

Conclusion and Future Work

This paper proposes a new tuning algorithm which minimizes the expected BLEU with ultraconservative update. By taking the progress made in previous rounds during the training process, our method obtains significant improvements over MERT on many test sets for both the Chinese-to-English and Spanish-to-English translation over the MOSES system. In future work, we will investigate our method on large training data.

Acknowledgments

We would like to thank Muyun Yang and Hongfei Jiang for many valuable discussions and thank three anonymous reviewers for many valuable comments and helpful suggestions. This work was supported by National Natural Science Foundation of China (61173073,61100093,61073130,61272384), the Key Project of the National High Technology Research and Development Program of China (2011AA01A207), and and the Fundamental Research Funds for Central Universites (HIT.NSRIF.2013065).

References

Chen, S. F. and Goodman, J. (1998). An empirical study of smoothing techniques for language modeling. In *Technical Report TR-10-98*. Harvard University.

Cherry, C. and Foster, G. (2012). Batch tuning strategies for statistical machine translation. In *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 427–436, Montréal, Canada. Association for Computational Linguistics.

Chiang, D., Marton, Y., and Resnik, P. (2008). Online large-margin training of syntactic and structural translation features. In *Proc. of EMNLP*. ACL.

Clark, J. H., Dyer, C., Lavie, A., and Smith, N. A. (2011). Better hypothesis testing for statistical machine translation: controlling for optimizer instability. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies: short papers - Volume 2*, HLT '11, pages 176–181, Stroudsburg, PA, USA. Association for Computational Linguistics.

Collins, M. (2002). Discriminative training methods for hidden markov models: Theory and experiments with perceptron algorithms. In *Proc. of EMNLP*. ACL.

Crammer, K., Dekel, O., Keshet, J., Shalev-Shwartz, S., and Singer, Y. (2006). Online passive-aggressive algorithms. *J. Mach. Learn. Res.*, 7:551–585.

Crammer, K. and Singer, Y. (2003). Ultraconservative online algorithms for multiclass problems. *J. Mach. Learn. Res.*, 3:951–991.

Horst, R. and Tuy, H. (1996). Global optimization: Deterministic approaches. Springer.

Koehn, P. (2004). Statistical significance tests for machine translation evaluation. In *Proc. of EMNLP*. ACL.

Koehn, P., Hoang, H., Birch, A., Callison-Burch, C., Federico, M., Bertoldi, N., Cowan, B., Shen, W., Moran, C., Zens, R., Dyer, C., Bojar, O., Constantin, A., and Herbst, E. (2007). Moses: open source toolkit for statistical machine translation. In *Proceedings of the 45th Annual Meeting of the ACL on Interactive Poster and Demonstration Sessions*, ACL '07, pages 177–180, Stroudsburg, PA, USA. Association for Computational Linguistics.

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

Kumar, S., Macherey, W., Dyer, C., and Och, F. (2009). Efficient minimum error rate training and minimum bayes-risk decoding for translation hypergraphs and lattices. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 1 - Volume 1*, ACL '09, pages 163–171, Stroudsburg, PA, USA. Association for Computational Linguistics.

Liu, L., Cao, H., Watanabe, T., Zhao, T., Yu, M., and Zhu, C. (2012). Locally training the log-linear model for smt. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 402–411, Jeju Island, Korea. Association for Computational Linguistics.

Macherey, W., Och, F. J., Thayer, I., and Uszkoreit, J. (2008). Lattice-based minimum error rate training for statistical machine translation. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, EMNLP '08, pages 725–734, Stroudsburg, PA, USA. Association for Computational Linguistics.

Och, F. J. (2003). Minimum error rate training in statistical machine translation. In *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics*, pages 160–167, Sapporo, Japan. Association for Computational Linguistics.

Och, F. J. and Ney, H. (2000). Improved statistical alignment models. In *Proceedings of the 38th Annual Meeting on Association for Computational Linguistics*, ACL '00, pages 440–447, Stroudsburg, PA, USA. Association for Computational Linguistics.

Papineni, K., Roukos, S., Ward, T., and Zhu, W.-J. (2002). Bleu: a method for automatic evaluation of machine translation. In *Proceedings of 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

Pauls, A., Denero, J., and Klein, D. (2009). Consensus training for consensus decoding in machine translation. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*, pages 1418–1427, Singapore. Association for Computational Linguistics.

Smith, D. A. and Eisner, J. (2006). Minimum risk annealing for training log-linear models. In *Proceedings of the COLING/ACL 2006 Main Conference Poster Sessions*, pages 787–794, Sydney, Australia. Association for Computational Linguistics.

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

Watanabe, T. (2012). Optimized online rank learning for machine translation. In *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 253–262, Montréal, Canada. Association for Computational Linguistics.

Watanabe, T., Suzuki, J., Tsukada, H., and Isozaki, H. (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, Prague, Czech Republic. Association for Computational Linguistics.

Zens, R., Hasan, S., and Ney, H. (2007). A systematic comparison of training criteria 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 524–532, Prague, Czech Republic. Association for Computational Linguistics.