A Corpus Level MIRA Tuning Strategy for Machine Translation

Ming Tan, Tian Xia, Shaojun Wang
Wright State University
3640 Colonel Glenn Hwy,
Dayton, OH 45435 USA
{tan.6, xia.7, shaojun.wang}
@wright.edu

Bowen Zhou
IBM T.J. Watson Research Center
1101 Kitchawan Rd,
Yorktown Heights, NY 10598 USA
zhou@us.ibm.com

Abstract

MIRA based tuning methods have been widely used in statistical machine translation (SMT) system with a large number of features. Since the corpus-level BLEU is not decomposable, these MIRA approaches usually define a variety of heuristic-driven sentence-level BLEUs in their model losses. Instead, we present a new MIRA method, which employs an exact corpus-level BLEU to compute the model loss. Our method is simpler in implementation. Experiments on Chinese-to-English translation show its effectiveness over two state-of-the-art MIRA implementations.

1 Introduction

Margin infused relaxed algorithm (MIRA) has been widely adopted for the parameter optimization in SMT with a large feature size (Watanabe et al., 2007; Chiang et al., 2008; Chiang et al., 2009; Chiang, 2012; Eidelman, 2012; Cherry and Foster, 2012). Since BLEU is defined on the corpus, and not decomposed into sentences, most MIRA approaches consider a variety of sentence-level BLEUs for the model losses, many of which are heuristic-driven (Watanabe et al., 2007; Chiang et al., 2008; Chiang et al., 2009; Chiang, 2012; Cherry and Foster, 2012). The sentence-level BLEU appearing in the objective is generally based on a pseudo-document, which may not precisely reflect the corpus-level BLEU. We believe that this mismatch could potentially harm the performance. To avoid the sentence BLEU, the work in (Haddow et al., 2011) proposed to process sentences in small batches. The authors adopted a Gibbs sampling (Arun et al., 2009) technique to search the *hope* and *fear* hypotheses, and they did not compare with MIRA. Watanabe (2012) also tuned the parameters with small batches of sentences and optimized a hinge loss not explicitly related to BLEU using stochastic gradient descent. Both approaches introduced additional complexities over baseline MIRA approaches.

In contrast, we propose a remarkably simple but efficient batch MIRA approach which exploits the exact corpus-level BLEU to compute model losses. We search for a *hope* and a *fear* hypotheses for the corpus with a straightforward approach and minimize the structured hinge loss defined on them. The experiments show that our method consistently outperforms two state-of-the-art MIRAs in Chinese-to-English translation tasks with a moderate margin.

2 Margin Infused Relaxed Algorithm

We optimize the model parameters based on *N*-best lists. Our development (*dev*) set is a set of triples $\{(f_i, \mathbf{e}_i, \mathbf{r}_i)\}_{i=1}^{M}$, where f_i is a source-language sentence, corresponded by a list of target-language hypotheses $\mathbf{e}_i = \{e_{ij}\}_{j=1}^{N(f_i)}$, with a number of references $\mathbf{r}_i$. $\mathbf{h}(e_{ij})$ is a feature vector. Generally, most decoders return a top-1 candidate as the translation result, such that $\bar{e}_i(\mathbf{w}) = \arg\max_j \mathbf{w} \cdot \mathbf{h}(e_{ij})$, where $\mathbf{w}$ are the model parameters. In this paper, we aim at optimizing the BLEU score (Papineni et al., 2002).

MIRA is an instance of online learning which assumes an overlap of the decoding procedure and the parameter optimization procedure. For example in (Crammer et al., 2006; Chiang et al., 2008), MIRA

Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing, pages 851–856,
Seattle, Washington, USA, 18-21 October 2013.

is performed after an input sentence are decoded, and the next sentence is decoded with the updated parameters. The objective for each sentence i is,

$$\min_{\mathbf{w}} \quad \frac{1}{2}||\mathbf{w}-\mathbf{w}'||^2 + C \cdot l_i(\mathbf{w}) \tag{1}$$

$$l_i(\mathbf{w}) = \max_{e_{ij}} \{\mathrm{b}(e_i^*) - \mathrm{b}(e_{ij}) - \mathbf{w}\cdot[\mathbf{h}(e_i^*) - \mathbf{h}(e_{ij})]\} \tag{2}$$

where $e_i^* \in \mathbf{e_i}$ is a *hope* candidate, $\mathbf{w}'$ is the parameter vector from the last sentence. Since MIRA defines its objective only based on the current sentence, $\mathrm{b}(\cdot)$ is a sentence-level BLEU.

Most MIRA algorithms need a deliberate definition of $\mathrm{b}(\cdot)$, since BLEU cannot be decomposed into sentences. The types of the sentence BLEU calculation includes: (a) a smoothed version of BLEU for e_{ij} (Liang et al., 2006), (b) fit e_{ij} into a pseudo-document considering the history (Chiang et al., 2008; Chiang, 2012), (c) use e_{ij} to replace the corresponding hypothesis in the oracles (Watanabe et al., 2007). The sentence-level BLEU sometimes perplexes the algorithms and results in a mismatch with the corpus-level BLEU.

3 Corpus-level MIRA

3.1 Algorithm

We propose a batch tuning strategy, corpus-level MIRA (c-MIRA), in which an objective is not built upon a hinge loss of a single sentence, but upon that of the entire corpus.

The online MIRAs are difficult to parallelize. Therefore, similar to the batch MIRA in (Cherry and Foster, 2012), we conduct the batch tuning by repeating the following steps: (a) Decode source sentences (in parallel) and obtain $\{\mathbf{e_i}\}_{i=1}^M$, (b) Merge $\{\mathbf{e_i}\}_{i=1}^M$ with the one from the previous iteration, (c) Invoke Algorithm 1.

We define $\mathcal{E} = (e_{\mathcal{E},1}, e_{\mathcal{E},2}, ..., e_{\mathcal{E},M})$ as a corpus hypothesis, with $\mathbf{H}(\mathcal{E}) = \frac{1}{M}\sum_{i=1}^{M}\mathbf{h}(e_{\mathcal{E},i})$. $e_{\mathcal{E},i}$ is the hypothesis of the source sentence f_i covered by $\mathcal{E}$. $\mathcal{E}$ is corresponded to a corpus-level BLEU, which we ultimately want to optimize. Following MIRA formulated in (Crammer et al., 2006; Chiang et al., 2008), c-MIRA repeatedly optimizes,

$$\min_{\mathbf{w}} \quad \frac{1}{2}||\mathbf{w}-\mathbf{w}'||^2 + C \cdot l_{corpus}(\mathbf{w}) \tag{3}$$

$$l_{corpus}(\mathbf{w}) = \max_{\mathcal{E}} \{\mathrm{B}(\mathcal{E}^*) - \mathrm{B}(\mathcal{E}) - \mathbf{w}\cdot[\mathbf{H}(\mathcal{E}^*) - \mathbf{H}(\mathcal{E})]\} \tag{4}$$

where $\mathrm{B}(\cdot)$ is a corpus-level BLEU. $\mathcal{E}^*$ is a *hope* hypothesis. $\mathcal{E} \in \mathcal{L}$, where $\mathcal{L}$ is the hypothesis space of the entire corpus, and $|\mathcal{L}| = |\mathbf{e_1}|\cdots|\mathbf{e_M}|$.

Algorithm 1 Corpus-Level MIRA

Require: $\{(f_i, \mathbf{e}_i, \mathbf{r}_i)\}_{i=1}^M$, $\mathbf{w_0}$, C

for $t = 1 \cdots T$ **do**
  $\mathcal{E}^* = \{\}$,$\mathcal{E}' = \{\}$ ▷ Initialize the *hope* and *fear*
  for $i = 1 \cdots M$ **do**
    $e_{\mathcal{E}^*,i} = \arg\max_{e_{ij}} [\mathbf{w_{t-1}} \cdot \mathbf{h}(e_{ij}) + \mathrm{b}'(e_{ij})]$
    $e_{\mathcal{E}',i} = \arg\max_{e_{ij}} [\mathbf{w_{t-1}} \cdot \mathbf{h}(e_{ij}) - \mathrm{b}'(e_{ij})]$
    $\mathcal{E}^* \leftarrow \mathcal{E}^* + \{e_{\mathcal{E}^*,i}\}$ ▷ Build the *hope*
    $\mathcal{E}' \leftarrow \mathcal{E}' + \{e_{\mathcal{E}',i}\}$ ▷ Build the *fear*
  end for
  $\triangle_{\mathrm{B}} = \mathrm{B}(\mathcal{E}^*) - \mathrm{B}(\mathcal{E}')$ ▷ the BLEU difference
  $\triangle_{\mathbf{H}} = \mathbf{H}(\mathcal{E}') - \mathbf{H}(\mathcal{E}^*)$ ▷ the feature difference
  $\alpha = \min\left[C, \frac{\triangle_{\mathrm{B}} + \mathbf{w_{t-1}}\cdot\triangle_{\mathbf{H}}}{||\triangle_{\mathbf{H}}||^2}\right]$
  $\mathbf{w_t} = \mathbf{w_{t-1}} - \alpha \cdot \triangle_{\mathbf{H}}$
  $\bar{\mathbf{w}}_t = \frac{1}{t+1}\sum_{t=0}^{t}\mathbf{w}_t$
end for
return $\bar{\mathbf{w}}_t$ with the optimal BLEU on the *dev* set.

c-MIRA can be regarded as a standard MIRA, in which there is only one single triple $(\mathcal{F}, \mathcal{L}, \mathcal{R})$, where $\mathcal{F}$ and $\mathcal{R}$ are the source and reference of the corpus respectively. Eq. 3 is equivalent to a quadratic programming with $|\mathcal{L}|$ constraints. Crammer et al. (2006) show that a single constraint with one *hope* $\mathcal{E}^*$ and one *fear* $\mathcal{E}'$ admits a closed-form update and performs well. We denote one execution of the outer loop as an *epoch*. The *hope* and *fear* are updated in each *epoch*. Similar to (Chiang et al., 2008), the *hope* and *fear* hypotheses are defined as following,

$$\mathcal{E}^* = \max_{\mathcal{E}}[\mathbf{w}\cdot\mathbf{H}(\mathcal{E}) + \mathrm{B}(\mathcal{E})] \tag{5}$$

$$\mathcal{E}' = \max_{\mathcal{E}}[\mathbf{w}\cdot\mathbf{H}(\mathcal{E}) - \mathrm{B}(\mathcal{E})] \tag{6}$$

Eq. 5 and 6 find the hypotheses with the best and worse BLEU that the decoder can easily achieve. It is unnecessary to search the entire space of $\mathcal{L}$ for precise solution $\mathcal{E}^*$and $\mathcal{E}'$, because MIRA only at-

tempts to separate the *hope* from the *fear* by a margin proportional to their BLEU differentials (Cherry and Foster, 2012). We just construct $\mathcal{E}^*$and $\mathcal{E}'$ respectively by,

$$e_{\mathcal{E}^*,i} = \max_{e_{i,j}}[\mathbf{w}\cdot\mathbf{h}(e_{i,j}) + \mathrm{b}'(e_{i,j})]$$
$$e_{\mathcal{E}',i} = \max_{e_{i,j}}[\mathbf{w}\cdot\mathbf{h}(e_{i,j}) - \mathrm{b}'(e_{i,j})]$$

where b' is simply a BLEU with add one smoothing (Lin and Och, 2004). A smoothed BLEU is good enough to pick up a "satisfying" pair of *hope* and *fear*. However, the updating step (Line 11) uses the corpus-level BLEU.

3.2 Justification

c-MIRA treats a corpus as one sentence for decoding, while conventional decoders process sentences one by one. We show the optimal solutions from the two methods are equivalent theoretically.

We follow the notations in (Och and Ney, 2002). We search a hypothesis on corpus $\mathcal{E} = \{e_{1,k_1}, e_{2,k_2}, ..., e_{M,k_M}\}$ with the highest probability given the source corpus $\mathcal{F} = \{\mathrm{f}_1, \mathrm{f}_2, ..., \mathrm{f}_\mathrm{M}\}$,

$$\begin{aligned}\overline{\mathcal{E}} &= \arg\max_{\mathcal{E}} \log P(\mathcal{E}|\mathcal{F}) \\ &= \arg\max_{\mathcal{E}} \left(\mathbf{w}\cdot\sum_{i=1}^{M}\mathbf{h}(e_{i,k_i}) - \sum_{\mathrm{i}=1}^{\mathrm{M}}\log(\mathrm{Z_i})\right) \quad (7) \\ &= \{\arg\max_{e_{i,k_i}} \mathbf{w}\cdot\mathbf{h}(e_{i,k_i})\}_{i=1}^{M} \quad (8)\end{aligned}$$

where $Z_i = \sum_{j=1}^{N(f_i)}\exp(\mathbf{w}\cdot\mathbf{h}(e_{i,j}))$, which is a constant with respective to $\mathcal{E}$. Eq. 7 shows that the feature vector of $\mathcal{E}$ is determined by the sum of each candidate's feature vectors. Also, the model score can be decomposed into each sentence in Eq. 8, which shows that decoding all sentences together equals to decoding one by one.

We also show that if the metric is decomposable, the loss in c-MIRA is actually the sum of the hinge loss $l_i(\mathbf{w})$ in structural SVM (Tsochantaridis et al., 2004; Cherry and Foster, 2012). We assume $B(e_{ij})$ to be the metric of a sentence hypothesis, then the loss of c-MIRA in Eq. 4 is,

$$\begin{aligned}l_{corpus}(\mathbf{w}) &\propto \max_{\mathcal{E}'}\sum_{i=1}^{M}[B(e_{i,k_{\mathcal{E}^*}}) - B(e_{i,k_{\mathcal{E}'}}) \\ &\quad -\mathbf{w}\cdot\mathbf{h}(e_{i,k_{\mathcal{E}^*}}) + \mathbf{w}\cdot\mathbf{h}(e_{i,k_{\mathcal{E}'}})] \\ &= \sum_{i=1}^{M}\max_{e_{ij}}[B(e_{i,k_{\mathcal{E}^*}}) - B(e_{ij}) \\ &\quad -\mathbf{w}\cdot\mathbf{h}(e_{i,k_{\mathcal{E}^*}}) + \mathbf{w}\cdot\mathbf{h}(e_{ij})] = \sum_{i=1}^{M} l_i(\mathbf{w})\end{aligned}$$

Instead of adopting a cutting-plane algorithm (Tsochantaridis et al., 2004), we optimize the same loss with a MIRA pattern in a simpler way. However, since BLEU is not decomposable, the structural SVM (Cherry and Foster, 2012) uses an interpolated sentence BLEU (Liang et al., 2006). Although Algorithm 1 has an outlook similar to the batch-MIRA algorithm in (Cherry and Foster, 2012), their loss definitions differ fundamentally. Batch MIRA basically uses a sentence-level loss, and they also follow the sentence-by-sentence tuning pattern. In the future work, we will compare structural SVM and c-MIRA under decomposable metrics like WER or SSER (Och and Ney, 2002).

4 Experiments and Analysis

We first evaluate c-MIRA in a iterative batch tuning procedure in a Chinese-to-English machine translation system with 228 features. Second, we show c-MIRA is also effective in the re-ranking task with more than 50,000 features.

In both experiments, we compare c-MIRA and three baselines: (1) MERT (Och, 2003), (2) Chiang et al.'s MIRA (MIRA_1) in (Chiang et al., 2008). (3) batch-MIRA (MIRA_2) in (Cherry and Foster, 2012). Here, we roughly choose C with the best BLEU on *dev* set, from $\{0.1, 0.01, 0.001, 0.0001, 0.00001\}$. We convert Chiang et al.'s MIRA to the batch mode described in section 3.1. So the only difference between MIRA_1 and MIRA_2 is: MIRA_1 obtains multiple constraints before optimization, while MIRA_2 only uses one constraint. We implement MERT and MIRA_1, and directly use MIRA_2 from Moses (Koehn et al., 2007). We conduct experiments in a server of 8-cores with 2.5GHz Opteron. We set the maximum number of *epochs* as we generally do not observe an obvious increase on the *dev* set BLEU.

		MERT	MIRA_1	MIRA_2	c-MIRA
	C		0.0001	0.001	0.0001
8	*dev*	**34.80**	34.70	34.73	34.70
feat.	04	**31.92**	31.81	31.73	31.83
	05	28.85	**28.94**	28.71	28.92
	C		0.001	0.001	0.001
all	*dev*	34.61	35.24	35.14	**35.56**
feat.	04	31.76	32.25	32.04	**32.57+**
	05	28.85	**29.43**	29.37	29.41
	06news	30.91	31.43	31.24	**31.82+**
	06others	27.43	28.01	28.13	**28.45**
	08news	25.62	26.11	26.03	**26.40**
	08others	16.22	16.66	16.46	**17.10+**

Table 1: BLEUs (%) on the *dev* and *test* sets with 8 dense features only and all features. The significant symbols (+ at 0.05 level) are compared with MIRA_2

The *epoch* size for MIRA_1 and MIRA_2 is 40, while the one for c-MIRA is 400. c-MIRA runs more *epochs*, because we update the parameters by much fewer times. However, we can implement Line 3$\sim$8 in Algorithm 1 in multi-thread (we use eight threads in the following experiments), which makes our algorithm much faster. Also, we increase the *epoch* sizes of MIRA_1 and MIRA_2 to 400, and find there is no improvement on their performance.

4.1 Iterative Batch Training

In this experiment, we conduct the batch tuning procedure shown in section 3. We align the FBIS data including about 230K sentence pairs with GIZA++ for extracting grammar, and train a 4-gram language model on the Xinhua portion of Gigaword corpus. A hierarchical phrase-based model (Chiang, 2007) is tuned on NIST MT 2002, which has 878 sentences, and tested on MT 2004, 2005, 2006, and 2008. All features used here, besides eight basic ones in (Chiang, 2007), consists of an extra 220 group features. We design such feature templates to group grammar by the length of source side and target side, $(feat\_type, a \leq src\_side \leq b, c \leq tgt\_side \leq d)$, where $feat\_type$ denotes any of relative frequency, reversed relative frequency, lexical probability and reversed lexical probability, and $[a, b]$, $[c, d]$ enumerate all possible subranges of $[1, 10]$, as the maximum length on each side of a hierarchical grammar is limited to 10. There are 4×55 extra group features. We also set the size of *N*-best list per sentence before merge as 200.

	MERT	MIRA_1	MIRA_2	c-MIRA
R. T.	25.8min	16.0min	7.3min	7.8min

Table 2: Running time.

All methods use 30 decoding iterations. We select the iteration with the best BLEU of the *dev* set for testing. We present the BLEU scores in Table 1 on two feature settings: (1) 8 basic features only, and (2) all 228 features. In the first case, due to the small feature size, MERT can get a better BLEU of the *dev* set, and all MIRA algorithms fails to generally beat MERT on the *test* set. However, as the feature size increase to 228, MERT degrades on the *dev*-set BLEU, and also become worse on *test* sets, while MIRA algorithms improve on the *dev* set expectedly. MIRA_1 performs better than MIRA_2, probably because of more constraints. c-MIRA can moderately improve BLEU by 0.2$\sim$0.4 from MIRA_1 and 0.2$\sim$0.6 from MIRA_2. This might indicate that a loss defined on corpus is more accurate than the one defined on sentence. Table 2 lists the running time. Only MIRA_2 is fairly faster than c-MIRA because of more *epochs* in c-MIRA.

4.2 Re-ranking Experiments

The baseline system is a state-of-the-art hierarchical phrase-based system, and trained on six million parallel sentences corpora available to the DARPA BOLT Chinese-English task. This system includes 51 dense features (including translation probabilities, provenance features, etc.) and about 50k sparse features (mostly lexical and fertility-based). The language model is a six-gram model trained on a 10 billion words monolingual corpus, including the English side of our parallel corpora plus other corpora such as Gigaword (LDC2011T07) and Google News. We use 1275 sentences for tuning and 1239 sentences for testing from the LDC2010E30 corpus respectively. There are four reference translations for each input sentence in both tuning and testing datasets.

We use a N-best list which is an intermediate out-

		MIRA_1	MIRA_2	c-MIRA
dense	dev	31.90	31.78	**32.00**
only	test	30.89	30.89	**31.07**
dense	dev	32.29	32.20	**32.49**
+sparse	test	31.12	31.00	**31.39**

Table 3: BLEUs (%) on re-ranking experiments.

MIRA_1	MIRA_2	c-MIRA
about 1,966,720	35,120	400

Table 4: Times of updating model parameters.

put of the baseline system optimized on TER-BLEU instead of BLEU. Before the re-ranking task, the initial BLEUs of the top-1 hypotheses on the tuning and testing set are 31.45 and 30.56. The average numbers of hypotheses per sentence are about 200 and 500, respectively for the tuning and testing sets. Again, we use the best *epoch* on the tuning set for testing. The BLEUs on *dev* and *test* sets are reported in Table 3. We observe that the effectiveness of c-MIRA is not harmed as the feature size is scaled up.

4.3 Analysis

To examine the simple search for *hopes* and *fears* (Line 3∼8 in Alg. 1), we use two *hope/fear* building strategies to get $\mathcal{E}^*$ and $\mathcal{E}'$: (1) simply connect each e_i^* and e_i' in Line 4∼5 of Algorithm 1, (2) conduct a slow beam search among the N-best lists of all foreign sentences from $\mathbf{e_1}$ to $\mathbf{e_M}$ and use Eq. 5 and 6 to prune the stack. The stack size is 10. We observe that there is no significant difference between the two strategies on the BLEU of the *dev* set. But the second strategy is about 10 times slower.

We also consider more constraints in Eq. 3. By beam search, we obtain one corpus-level oracle and 29 other hypotheses similar to (Chiang et al., 2008), and optimize with SMO (Platt, 1998). Unfortunately, experiments show that more constraints lead to an overfitting and no improved performance.

As shown in Table 4, in one execution, our method updates the parameters by only 400 times; MIRA_2 updates by $40 \times 878 = 35120$ times; and MIRA_1 updates much more (about 1,966,720 times) due to the SMO procedure. We are surprised to find c-MIRA gets a higher training BLEU with such few parameter updates. This probably suggests that there is a gap between sentence-level BLEU and corpus-level BLEU, so standard MIRAs need to update the parameters more often.

Regarding simplicity, MIRA_1 uses a strongly-heuristic definition of a sentence BLEU, and MIRA_2 needs a pseudo-document with a decay rate of $\gamma = 0.9$. In comparison, c-MIRA avoids both the sentence level BLEU and the pseudo-document, thus needs fewer variables.

5 Conclusion

We present a simple and effective MIRA batch tuning algorithm without the heuristic-driven calculation of sentence-level BLEU, due to the indecomposability of a corpus-level BLEU. Our optimization objective is directly defined on the corpus-level hypotheses. This work simplifies the tuning process, and avoid the mismatch between the sentence-level BLEU and the corpus-level BLEU. This strategy can be potentially applied to other optimization paradigms, such as the structural SVM (Cherry and Foster, 2012), SGD and AROW (Chiang, 2012), and other forms of samples, such as forests (Chiang, 2012) and lattice (Cherry and Foster, 2012).

6 Acknowledgments

The key idea and a part of the experimental work of this paper were developed in collaboration with the IBM researcher when the first author was an intern at IBM T.J. Watson Research Center. This research is partially supported by Air Force Office of Scientific Research under grant FA9550-10-1-0335, the National Science Foundation under grant IIS RI-small 1218863 and a Google research award.

References

A. Arun, C. Dyer, B. Haddow, P. Blunsom, A. Lopez, and P. Koehn. 2009. Monte Carlo inference and maximization for phrase-based translation. *In Proceedings of the Thirteenth Conference on Computational Natural Language Learning (CoNLL)*, 102-110.

C. Cherry and G. Foster. 2012. Batch tuning strategies for statistical machine translation. *Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, 427-436.

D. Chiang. 2012. Hope and fear for discriminative training of statistical translation models. *Journal of Machine Learning Research (JMLR)*, 1159-1187.

D. Chiang, K. Knight, and W. Wang. 2009. 11,001 new features for statistical machine translation. *Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, 218-226.

D. Chiang, Y. Marton, and P. Resnik. 2008. Online large-margin training of syntactic and structural translation features. *In Proc. of Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 224-233.

D. Chiang. 2007. Hierarchical phrase-based translation. *Computational Linguistics*, 33(2):201-228.

K. Crammer, O. Dekel, J. Keshet, S. Shalev-Shwartz, and Y. Singer. 2006. Online passive-aggressive algorithms. *Journal of Machine Learning Research (JMLR)*, 7:551-585.

V. Eidelman. 2012. Optimization strategies for online large-margin learning in machine translation. *Proceedings of the Seventh Workshop on Statistical Machine Translation*, 480-489.

B. Haddow, A. Arun, and P. Koehn. 2011. SampleRank training for phrase-based machine translation. *Proceedings of the Sixth Workshop on Statistical Machine Translation. Association for Computational Linguistics*, 261-271.

P. Koehn, H. Hoang, A. Birch, C. Burch, M. Federico, N. Bertoldi, B. Cowan, W. Shen, C. Moran, R. Zens, C. Dyer, O. Bojar, A. Constantin, and E. Herbst. 2007. Moses: Open source toolkit for statistical machine translation. *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*, 177-180.

P. Liang, A. Bouchard-Cote, D. Klein, and B. Taskar. 2006. An end-to-end discriminative approach to machine translation. *In Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics*, 761-768.

C. Lin and F. Och. 2004. Orange: a method for evaluating automatic evaluation metrics for machine translation. *In Proc. of International Conference on Computational Linguistics (COLING)*, No. 501.

F. Och. 2003. Minimum error rate training in statistical machine translation. *Proceedings of the 41st Annual Meeting on Association for Computational Linguistics (ACL)*, 160-167.

F. Och and H. Ney. 2002. Discriminative training and maximum entropy models for statistical machine translation. *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics (ACL)*, 295-302.

K. Papineni, S. Roukos, T. Ward, and W. Zhu. 2002. BLEU: a method for automatic evaluation of machine translation. *Proceedings of the 40th annual meeting on association for computational linguistics. Association for Computational Linguistics (ACL)*, 311-318.

J. Platt. 1998. Sequetial minimal optimization: A fast algorithm for training support vector machines. *In Technical Report MST-TR-98-14. Microsoft Research*.

I. Tsochantaridis, T. Hofman, T. Joachims, and Y. Altun. 2004. Support vector machine learning for interdependent and structured output spaces. *International Conference on Machine Learning (ICML)*, 823-830.

T. Watanabe. 2012. Optimized online rank learning for machine translation. *Proceedings of Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, 253-262.

T. Watanabe, J. Suzuki, H. Tsukada, and H. Isozaki. 2007. Online large-margin training for statistical machine translation. *Proceedings of Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, 764-773.