Maximum Rank Correlation Training for Statistical Machine Translation

Daqi Zheng **Yifan He**† **Yang Liu** **Qun Liu**
Key Laboratory of Intelligent Information Processiong
Institute of Computing Technology, Chinese Academy of Sciences
{zhengdaqi, yliu, liuqun}@ict.ac.cn
† CNGL, School of Computing, Dublin City University
yhe@computing.dcu.ie

Abstract

We propose Maximum Ranking Correlation (MRC) as an objective function in discriminative tuning of parameters in a linear model of Statistical Machine Translation (SMT). We try to maximize the ranking correlation between sentence level BLEU (SBLEU) scores and model scores of the N-best list, while the MERT paradigm focuses on the potential 1-best candidates of the N-best list. After we optimize the MER and the MRC objectives using an multiple objective optimization algorithm at the same time, we interpolate them to obtain parameters which outperform both. Experimental results on WMT French–English data set confirm that our method significantly outperforms MERT on out-of-domain data sets, and performs marginally better than MERT on in-domain data sets, which validates the usefulness of MRC on both domain specific and general domain data.

1 Introduction

Searching for the optimal parameters in linear models (Och and Ney, 2002) of Statistical Machine Translation (SMT) has been a major challenge to the MT community. The most widely used approach to-date is Minimum-Error-Rate Training (MERT:(Och, 2003)), which tries to find the parameters that optimize the translation quality of the 1-best translation candidate, using the N-best list as an approximation of the decoder's search space.

In spite of its usefulness and high adoption, MERT suffers from shortcomings that the MT community is becoming aware of. On the one hand,

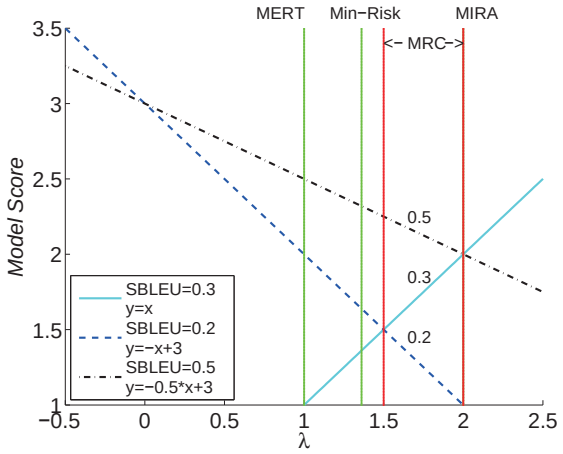

Figure 1: Made-up Example: The 3 sloping lines represent all 3 candidates in N-best list. Their SBLEU ($BLEU = SBLEU$ when only one sentence) and funtions are in the legend of figure.

MERT is not designed for models with rich features and therefore leads to translations of unstable quality in such scenarios. The fluctuation in quality can even be statistically perceivable when the number of features is larger than 25 or 30 in practice; on the other hand, Smith (2006) finds that, MERT relies heavily on the behavior of parameters on the error surface, which is likely to be affected by random variances in the N-best list, and also lead to less generalizable results especially when the development set and the test set are not from exactly the same domain.

Both the former and the latter shortcomings have been studied in recent research. e.g. The Margin Infused Relaxed Algorithm (MIRA: (Chiang et al.,

2008; Chiang et al., 2009)) is shown to be capable of handling tens of thousands of features in training, while Cer et. al (2008) try to overcome irregularities on the error surface of MERT.

In this paper, we focus on improving the generalizability of MERT by introduce a new objective function MRC, which restricts the permutation of the whole N-best list. In the MERT paradigm, the tuning objective is based on the 1-best error surface of the N-best list of an in-domain test set. As MERT actually optimizes parameters for the 1-best for a particular domain, the resulting parameters become domain specific, and does not generalize well across domains.

However, in real world translation tasks, it is not uncommon that people have to translate content from different domains. Therefore, we propose to add more optimization objectives to MT tuning so as to improve the generalizability of the resulting parameters. As a first step, we add ranking correlation to the objective, and find this can bring 0.4 improvement in terms of the BLEU score in a cross-domain translation task. To make this optimization practical, we use non-dominated sorting genetic algorithm II (NSGA-II) (Deb et al., 2002), a genetic algorithm to perform multi-objective optimization.

For example, in Figure 1, MERT chooses the middle point of two cross points. By contrast, MRC tries to maximize the rank correlation between SBLEU and the model score. and will adjust λ into the open interval $(1.5, 2)$, in which the order of candidates' model score is perfectly the same as their SBLEU. We obtain $\lambda = 1.37$ via assuming the objective of Min-Risk is the expectation of BLEU, and the probability of each (c)andicate is given by $p(c_i) = \exp(\gamma \cdot score(c)) / \sum_i \exp(\gamma \cdot score(c_i))$ with $\gamma = 1$ (Li and Eisner, 2009). In MIRA (Chiang et al., 2008), if we choose candidates whose SBLEU are 0.5 and 0.2 as positive and negative examples respectively, MIRA will make the margin between them as large as possible and λ will no smaller than 2.

2 Related Work

Many people have tried to improve MERT in different aspects, such as to improve its stability (Foster and Kuhn, 2009), to improve its performance (Duh and Kirchhoff, 2008), to extent the search space (Macherey et al., 2008; Kumar et al., 2009; Chatterjee and Cancedda, 2010), and to improve the optimization algorithm itself (Lambert and Banchs, 2006; Cer et al., 2008; González-Rubio et al., 2009). Some even replace it completely (Turian et al., 2007; Blunsom et al., 2008). Some people try other objectives during the decoding phase (Kumar and Byrne, 2004; Tromble et al., 2008; Li et al., 2009; DeNero et al., 2009), while others change it in training phase (Zens et al., 2007; He and Way, 2009; He and Way, 2010). There is research that introduces other objectives during tuning (Chiang et al., 2008; Li and Eisner, 2009; Pauls et al., 2009; Hopkins and May, 2011), but these objectives are different from the MRC objective presented in this paper.

3 Maximum Rank Correlation Training

3.1 The Training Paradigm

Using the N-best derivation of a decoder to approximate its search space, we find the optimal set of parameters $\hat{\lambda}$ (Fig 1), which maximizes the weighted sum of correlation on a set with M sentences, as in Eq. (1).

$$\hat{\lambda} = \arg\max_{\lambda}(\sum_{i=1}^{M} w_i \cdot Corr_i(\lambda)) \quad (1)$$

where w_i is the weight of the i–th sentence, and $Corr_i(\lambda)$ is the correlation between the model scores and the translation quality of the translation candidates approximated by SBLEU, as in Eq. (2).

$$Corr_i(\lambda) = Corr(\Phi_1^N(\lambda), SBLEU(\mathbf{e}_1^N))) \quad (2)$$

where $\mathbf{e}$ is the i–th sentence, $\mathbf{e}_1^N$ is the N-best derivation of the decoder and $\Phi_1^N(\lambda)$ are the model scores for $\mathbf{e}_1^N$ using parameters λ. We calculate SBLEU by applying the BLEU (Papineni et al., 2002) formulation directly to single sentence.

There exist many coefficients to measure the correlation between model scores and SBLEU scores. In our implementation, we use the Spearman's ρ ranking correlation, as in Eq. (3)

$$\rho = \frac{\sum_i (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_i (x_i - \bar{x})^2 \sum_i (y_i - \bar{y})^2}} \quad (3)$$

where x_i and y_i are respectively the rank of the model score and the rank of the SBLEU score of the i-th derivation in N-best; $\bar{x}$ and $\bar{y}$ are the means of the rank of the scores. It can be considered as the Pearson correlation coefficient between the ranked variables. We choose Spearman's rank correlation coefficient because we are not interest in the values of model scores, but in whether the parameters λ is able to produce the most similar ranking compared to the SBLEU gold standard. Therefore we use ρ to calculate $Corr$ in Eq. (2). This is similar to what people do when comparing auto evaluation metric scores with human judgements. We focus on ranks instead of the values of scores to relax the linear constraint introduced by the Pearson correlation.

3.2 Combination of MRC and MER Training

Inspired by Chiang et al. (2008), we also explore the possibility of combining the rank correlation with evaluation metric score as an alternative to the MER and the MRC objectives. Given a set of features λ, we perform a straightforward linear combination as in Eq. (4)

$$\hat{\lambda} = \arg\max_{\lambda}(\alpha \cdot \sum_{i=1}^{M} Corr_i(\lambda) + (1-\alpha) \cdot BLEU(\lambda)) \quad (4)$$

where $BLEU(\lambda)$ is the system level BLEU score on the development set using λ, and α is the interpolation parameter. The result is shown in Fig 3.

Note that this combination can also be viewed as using ranking correlation as the regularization for the single top-1 BLEU.

3.3 Optimization

As we use the non-parametric Spearman's correlation ρ as the optimization objective, we do not have an analytic method such as the gradient decent algorithm to optimize the objective directly. More importantly, if we try to replace BLEU with Spearman's ρ in MERT, there is no convex hull of 1–best changes, we have to re-calculate ρ each time when one candidate in the N–best list changes its rank. To make the optimization practical, we rely on a genetic algorithm to find the solution.

Furthermore, as we want to test the combination of different objectives, we have to perform multi-objective optimization to avoid the overhead of tuning on each combination separately. For this task, multi-objective evolutionary algorithms (MOEA) (Fonseca et al., 1993) have been successfully used in natural language processing (NLP) areas such as word alignment (Chen et al., 2009), parsing, and tagging (Araujo, 2006).

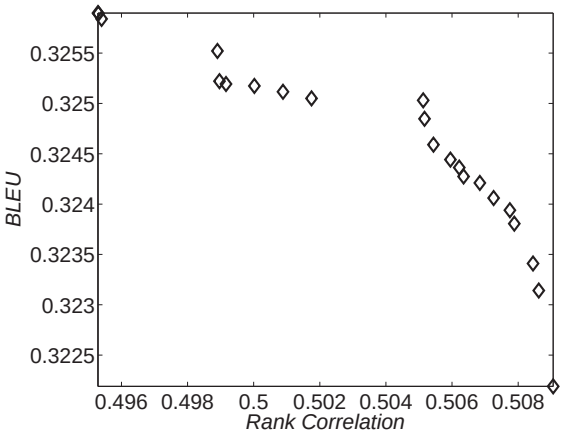

Figure 2: Pareto optimum set on the development set: parameters which have better correlation always mean worse BLEU, but often achieve better BLEU on test set. By definition of Pareto-optimum, we cannot find parameters that achieve both better correlation and better BLEU score than the Pareto-optimal feature sets.

MOEAs are based on the concept of the Pareto-optimal set, in which no solution can achieve better score than any other in every objective. The biggest advantage of MOEA is that it has the ability to find multiple Pareto-optimal solutions (Fig 2) in one single simulation run, instead of targeting different combinations of these objectives at each run. In this paper, we choose an effective MOEA: NSGA-II, which can produce a diverse set of solutions by forcing the solutions in Pareto-optimal set to keep distance from each other, and at the same time encouraging them to move toward the true Pareto-optimal set.

4 Experiments

In order to compare our paradigm with MERT, we perform parameter tuning on the same model using both MERT and our proposed method.

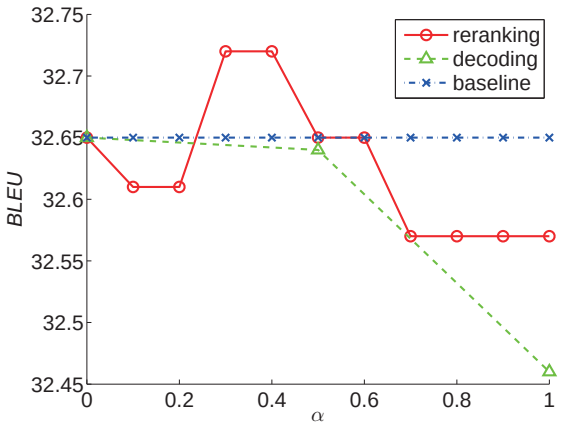

Figure 3: Test05: BLEU varies with different α in Eq. 4 to interpolate between MRC and MER. From this figure, we choose $\alpha = 0.4$ as it seems performs best on Test05.

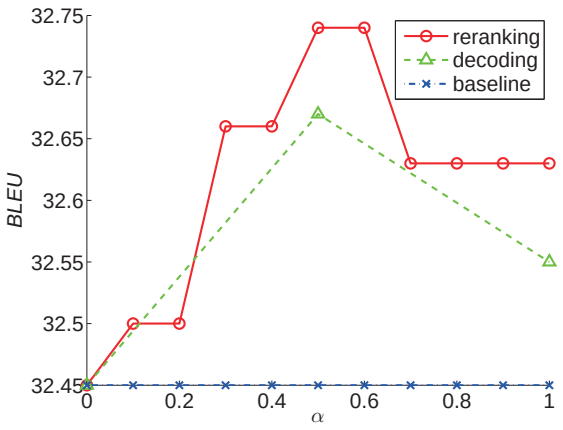

Figure 4: Test06: Another in-domain test set to demonstrate the advantage of interpolation of objectives

4.1 Experimental setting

4.1.1 Data

We use the French–English parallel data provided by the WMT08 [1] shared translation task. The training data is the Europarl v3b release (Koehn, 2005). The language model corpus is the English part of monolingual language model training data provided by the organizers of WMT

From the data sets provided by WMT, we use `dev2006` as the tuning set for λ, `test2005` as tuning set for α, and `test2006`, `test2007` and `test2008` as in-domain test sets. We use `newstest2008`, `newstest2009` and `newstest2010` as out-of-domain test sets.

[1] `http://www.statmt.org/wmt08`

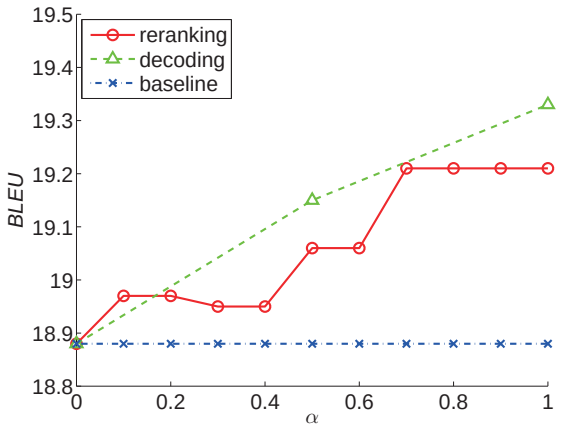

Figure 5: Newstest08: On the Out-of-Domain test set, MRC not only performs better than MER, but also beat the interpolation's perfomance. Notice: the directly decoding method outperform the reranking one.

4.1.2 Program

We use Moses [2] as the baseline decoder, perform word alignment using the GIZA++ (Och and Ney, 2003) implementation of IBM Model 4, and extract phrases using the grow-diag-final heuristic (Koehn et al., 2003). We train a 4-gram language model using the SRILM language modeling toolkit (Stolcke, 2002). We use the MERT implemented by Bertoldi et al., (2009), which is included in the Moses package. BLEU is calculated by the mteval script provided by NIST [3]. Statistical significance of test results is computed by Koehn's boosting tool (Koehn, 2004).

We preprocess the data using the toolkits provided by WMT08 organizers, train and tune Moses following the WMT baseline description, with 100-best list, using up to 15 tuning iterations, and finally arriving at a model with 14 default features.

4.2 The Baseline System

We tune 3 times on the tuning set and pick the experiment whose parameters achieve the highest BLEU score on the `test05` as a baseline. We then use this parameter to decode all other sets, and obtain results

[2] checkout from svn with version 3625

[3] `ftp://jaguar.ncsl.nist.gov/mt/resources/mteval-v11b.pl`

indicated by the dash-dot line in Fig 3, Fig 4 and Fig 5

4.3 The Proposed System

4.3.1 Spearman ρ

We use a part of goose[4] to calculate the Spearman's rank correlation coefficient. [5]

The weight in Eq. (1) of each sentence in the tuning set is set to be

$$w_i = length(\mathbf{f}_i) / \sum_{i=1}^{M} (length(\mathbf{f}_i)) \qquad (5)$$

where $\mathbf{f}_i$ is the i-th sentence.

4.3.2 Genetic Algorithm

To utilize the parameters produced by MERT, we change NSGA-II to allow seeding the first generation of the algorithm with user-defined parameters. We use two real value objectives in NSGA–II, so only the real part of the parameters need to be set. Following the given example in the software package, we set interval of variables to [-1, 1], probability of crossover to 0.9, probability of mutation to 0.5, distribution index for real variable SBX crossover to 10, and distribution index for real variable polynomial mutation to 20. Additionally, we set the random seed of the algorithm to 0.2 to make the experiments repeatable.

4.3.3 Procedure

We collect all parameters generated at each round by MERT in baseline, (10 in this case), and use them to initialize 10 individuals in the first generation of the NSGA-II algorithm, and randomize the remaining 390 individuals. We run the algorithm for 100 generations, each generation with 400 individuals. The algorithm uses the same input data as the last round of MERT, which is the N-best generated at each round of the baseline. The difference is that instead of only outputing one parameter like MERT, the algorithm outputs a set of parameters. Fig 2 illustrates the Pareto frontier we obtain from the NSGA-II optimization on `test2005`, which consists of Pareto-optimal feature sets.

[4] `http://www.gnu.org/software/goose/`

[5] We fixed a bug by ourselves in this software and plan to release the fixed code later, as the software is discontinued.

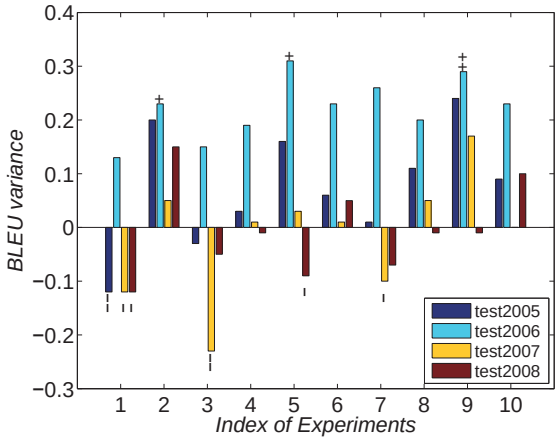

Figure 6: In-Domain: The improvement on each single experiment. The variation of baselines comes from the randomization inside newest MERT. Here and below, + or ++ indicates significantly better than MERT baseline ($p < 0.05$ or $p < 0.01$, respectively), - or -- indicate significantly worse than MERT baseline ($p < 0.05$ or $p < 0.01$, respectively)

We optimize on both MER and the MRC, so the output of the optimization will be the Pareto-optimal set of λ.

4.3.4 Reranking and Decoding

We have to find the λ by interpolating the BLEU score and the ranking correlation score.

We perform interpolation on the Pareto frontier and try to find the best α and experiment with the method proposed in this paper in two settings.

Firstly, we use MRCT in the *reranking* setting. In *reranking* we rerank the baseline's N-best of test sets using the interpolated parameters, where α is in [0.0, 1.0] with the step size 0.1. Fig 3, Fig 4 and 5 illustrate the results in red solid line.

We also experiment with the *decoding* setting, where we use the interpolated parameters to decode the test set directly, without reranking. As decoding requires more resource than *reranking*, we only interpolate with $\alpha = 0.0, 0.5$, and 1.0. Fig 3, Fig 4 and 5 illustrate the results in green dashed line.

4.4 Avoiding Random Noise

In order to test whether the improvements in BLEU score are resulted in the randomization in MERT, we rerun the reranking method from MERT's tuning to the end 10 times (Here in each experiment, we just

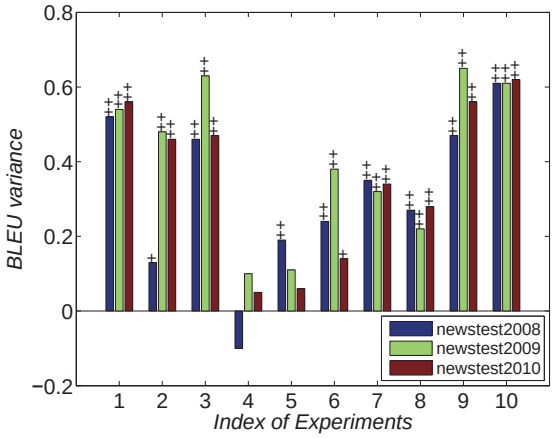

Figure 7: Out-Domain: The figure demonstrates the significant improvements on Out-of-Domain set

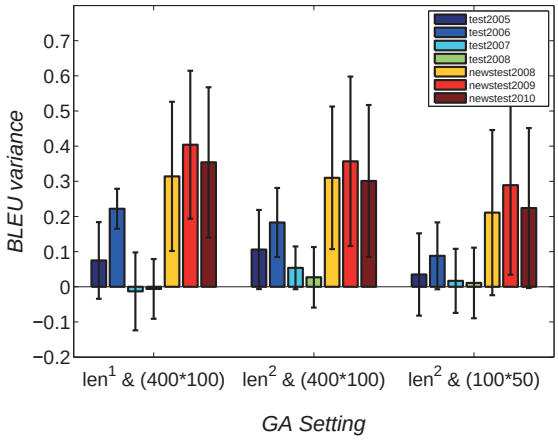

Figure 8: Mean & STD: The mean and standard deviation of BLEU score variance on different settings. $len\&(400 \cdot 100)$: len indicates $length()$ in Eq. (5). $(400 \cdot 100)$ are $(population \cdot generation)$ setting in NSGA-II. Improvement is more stable and significant on Out-of-Domain than In-Domain ones.

run 1 instead of 3 times the baseline system, which is different from what we do in Section 4.2 Sec 4.2), indexed from 1 to 10. We set $\alpha = 0.4$ in accordance with the results showed in Fig 3. We show the variances of each run in Fig 6 and Fig 7. We run the experiment from MERT's tuning to the *reranking* 10 times.

We also try different setting of sentence weight and genetic algorithm, the results are reported in Fig 8.

5 Analysis of Results

We report results in Fig 4, Fig 5, Fig 6 and Fig 7, then summarize the results in Fig 8. The first 4 bars in each group are the results on the in-domain test sets, and the following 3 higher bars are the results on the out-of-domain test sets.

In this case, the leftmost point & dash-dot line in the Fig 3, Fig 4 and Fig 5 where $\alpha = 0$ actually equals to the baseline system where only MERT is used to find the parameters. That is because the MER parameter set found by MERT is still kept in the Pareto-frontier determined by the NSGA-II algorithm, which confirms MERT's ability to find an optimal feature set $\hat{\lambda}$ on the *tuning* set.

5.1 Performance on In-Domain Data

We first look at the performance of our method on the in-domain test data. As shown in Figures 4 and 6, the reranking technique steadily outperforms the decoding technique. We note that the ranking method generally performs better than the other configurations in all data sets. The difference is more clear using α around 0.4, and is statistically significant on `test2006`.

5.2 Performance on Out-of-Domain Data

On the out-of-domain data set, the improvement brought by our proposed method is more perceivable. Except for very few exceptions using the reranking technique, both techniques can consistently outperform the baseline which uses MERT. As shown in Figure 7, the difference is often statistically significant.

Looking at Figure 5, we find that between the ranking and the decoding method, the decoding performs much better. We suspect that the parameters favored by the MER overfits the development set, and this undesired tendency can be very much alleviated by the introduction of more generalizable tuning objectives, such as MRC.

Results on the out-of-domain data sets, shown in Figure 7, also confirm the necessity of combining multiple tuning objectives in MT tuning.

5.3 Performance Comparison

Comparing the performance of our method on the in-domain and out-of-domain data sets, we find that MRCT performs much better on the out-of-domain

Index of Exps	1	2	3	4	5	6	7	8	9	10
# of Iteration	11	15	15	12	9	10	15	12	14	15
Last Mert	26	22	28	23	17	13	23	12	18	26
Tuning	1222	1766	1657	1329	1047	1113	1694	1363	1613	1741
$len^1\&(400 \cdot 100)$	780	615	782	793	570	448	755	499	606	793
$len^2\&(400 \cdot 100)$	768	624	769	778	569	431	718	499	607	741
$len^2\&(100 \cdot 50)$	95	76	96	97	70	55	89	63	75	96

Table 1: 10 experiment's Running Time: in 100 seconds. Compare the GA with the total tuning time, and consider it need only run once at the tuning phase, the computation cost is affordable.

data[6], and can still help the performance when combined with other tuning objectives on the in-domain data.

Putting Fig 8 and Table 1 together, we can find that, in the $(400 \cdot 100)$ setting, MRC generally performs better than MER, and MRC can still obtain steady improvements in far less time with the $(100 \cdot 50)$ setting. Compare to the total tuning time, even the time taken by the $(400 \cdot 100)$ setting is rational, for the NSGA-II is only run once at training time.

These results demonstrate that our method can bring steady improvement for cross-domain translation. Given the fact that purely in-domain data is rarely found in the real world use cases, our method's ability to generalize to unknown domains is desirable in real world translation tasks.

6 Conclusion

6.1 Conclusions and Future Work

In this paper we presented Maximum Ranking Correlation Training (MRCT) for tuning MT systems which was different from the MER. We optimized to maximize the correlation between the model scores and the BLEU scores on the N-best output of the decoder, and improved the robustness of our method by combining this MRCT with MERT in an evolution algorithm framework. We performed experiments on both the in-domain and the out-of-domain data set. Experimental results confirm that our method consistently outperforms MERT on out-of-domain data sets, and is on-par or slightly better than MERT on in-domain data sets.

[6]Li et al. (2009), Pauls et al. (2009) and Chiang(2008) also reports similar results although Chiang(2008) does not described them explicitly

The most important characteristic of our method is that it is easily extensible. We therefore plan to experiment with more new optimizing objectives and other optimizing algorithms, to exploit more features in translation, and to extend our method to other formalisms, such as hierachical (Chiang, 2005; Chiang, 2007) or syntax-based (Galley et al., 2006; Liu et al., 2006) translation.

7 Acknowledgement

Daqi Zheng, Yang Liu and Qun Liu are supported by National Natural Science Foundation of China Contract 60903138, 90920004, and 60736014. Yifan He is supported by Science Foundation Ireland (Grant No 07/CE/I1142). This work is part funded under FP7 of the EC within the EuroMatrix+ project (Grant No 231720). We thank Jennifer Foster and Joachim Wagner for their valuable suggestions.

References

L. Araujo. 2006. Multiobjective genetic programming for natural language parsing and tagging. *Parallel Problem Solving from Nature-PPSN IX*, pages 433–442.

N. Bertoldi, B. Haddow, and J.B. Fouet. 2009. Improved Minimum Error Rate Training in Moses. *The Prague Bulletin of Mathematical Linguistics*, 91(-1):7–16.

P. Blunsom, T. Cohn, and M. Osborne. 2008. A discriminative latent variable model for statistical machine translation. *Proc. of ACL-HLT*.

D Cer, D Jurafsky, and C Manning. 2008. Regularization and search for minimum error rate training. In *Proc. of SMT*, pages 26–34, Columbus, OH.

S. Chatterjee and N. Cancedda. 2010. Minimum error rate training by sampling the translation lattice. In *Proc. of EMNLP*, pages 606–615. ACL.

Yidong Chen, Xiaodong Shi, Changle Zhou, and Qingyang Hong. 2009. A word alignment

model based on multiobjective evolutionary algorithms. *Computers & Mathematics with Applications*, 57(11-12):1724 – 1729. Proceedings of the International Conference, Bio-Inspired Computing-Theories and Applications BIC-TA 2007 Zhengzhou, China.

D. Chiang, Y. Marton, and P. Resnik. 2008. Online large-margin training of syntactic and structural translation features. In *Proc. of EMNLP*, pages 224–233. ACL.

D. Chiang, K. Knight, and W. Wang. 2009. 11,001 new features for statistical machine translation. In *Proc. of EMNLP*, pages 218–226. ACL.

D Chiang. 2005. A hierarchical phrase-based model for statistical machine translation. In *Proc. of ACL*, pages 263–270, Ann Arbor, Michigan.

D Chiang. 2007. Hierarchical phrase-based translation. *Comput. Linguist.*, 33(2):201–228.

K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan. 2002. A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE transactions on evolutionary computation*, 6(2):182–197.

J. DeNero, D. Chiang, and K. Knight. 2009. Fast consensus decoding over translation forests. In *Proc. of ACL-AFNLP*, pages 567–575. ACL.

K. Duh and K. Kirchhoff. 2008. Beyond log-linear models: boosted minimum error rate training for N-best Re-ranking. In *Proc. ACL-HLP*, pages 37–40. ACL.

C.M. Fonseca, P.J. Fleming, et al. 1993. Genetic algorithms for multiobjective optimization: Formulation, discussion and generalization. In *Proceedings of the fifth international conference on genetic algorithms*, volume 423, pages 416–423. Citeseer.

G. Foster and R. Kuhn. 2009. Stabilizing minimum error rate training. In *Proc. of SMT*, pages 242–249. ACL.

M Galley, J Graehl, K Knight, D Marcu, S DeNeefe, W Wang, and I Thayer. 2006. Scalable inference and training of context-rich syntactic translation models. In *Proc. of COLING-ACL*, pages 961–968.

J. González-Rubio, D. Ortiz-Martinez, and F. Casacuberta. 2009. Minimum error-rate training in statistical machine translation using SVMs.

Y. He and A. Way. 2009. Improving the objective function in minimum error rate training. *Proc. of MT summit*, pages 238–245.

Y. He and A. Way. 2010. Metric and reference factors in minimum error rate training. *Machine Translation*, pages 1–12.

M. Hopkins and J. May. 2011. Tuning as ranking. In *Proc. of EMNLP*.

P Koehn, F J. Och, and D Marcu. 2003. Statistical phrase-based translation. In *Proc. of HLT-NAACL*, pages 127–133, Edmonton, Canada.

P. Koehn. 2004. Statistical significance tests for machine translation evaluation. In *Proc. of EMNLP*, volume 4, pages 388–395.

P. Koehn. 2005. Europarl: A parallel corpus for statistical machine translation. In *MT summit*, volume 5.

S. Kumar and W. Byrne. 2004. Minimum bayes-risk decoding for statistical machine translation. In *Proc. of HLT-NAACL*, pages 169–176.

S. Kumar, W. Macherey, C. Dyer, and F. Och. 2009. Efficient minimum error rate training and minimum bayes-risk decoding for translation hypergraphs and lattices. In *Proc. of ACL-AFNLP*, pages 163–171. Association for Computational Linguistics.

P. Lambert and R.E. Banchs. 2006. Tuning machine translation parameters with SPSA. In *Proc. of IWSLT*, pages 190–196. Citeseer.

Z. Li and J. Eisner. 2009. First-and second-order expectation semirings with applications to minimum-risk training on translation forests. In *Proc. of EMNLP*, pages 40–51. ACL.

Z. Li, J. Eisner, and S. Khudanpur. 2009. Variational decoding for statistical machine translation. In *Proc. of ACL-AFNLP*, pages 593–601. ACL.

Y Liu, Q Liu, and S Lin. 2006. Tree-to-string alignment template for statistical machine translation. In *Proc. of COLING-ACL*, pages 609–616.

W. Macherey, F.J. Och, I. Thayer, and J. Uszkoreit. 2008. Lattice-based minimum error rate training for statistical machine translation. In *Proc. of EMNLP*, pages 725–734. ACL.

F J. Och and H Ney. 2002. Discriminative training and maximum entropy models for statistical machine translation. In *Proc. of ACL*, pages 295–302.

F J. Och and H Ney. 2003. A systematic comparison of various statistical alignment models. *Computational Linguistics*, 29(1):19–51.

F J. Och. 2003. Minimum error rate training in statistical machine translation. In *Proc. of ACL*, pages 160–167.

K. Papineni, S. Roukos, T. Ward, and W.J. Zhu. 2002. BLEU: a method for automatic evaluation of machine translation. In *Proc. of ACL*, pages 311–318. ACL.

A. Pauls, J. DeNero, and D. Klein. 2009. Consensus training for consensus decoding in machine translation. In *Proc. of EMNLP*, pages 1418–1427. ACL.

D.A. Smith and J. Eisner. 2006. Minimum risk annealing for training log-linear models. In *Proc. of the COLING/ACL on Main conference poster sessions*, pages 787–794. ACL.

A Stolcke. 2002. Srilm-an extensible language modeling toolkit. In *Proc. of ICSLP*, volume 30, pages 901–904.

R.W. Tromble, S. Kumar, F. Och, and W. Macherey. 2008. Lattice Minimum Bayes-Risk decoding for statistical machine translation. In *Proc. of EMNLP*, pages 620–629. ACL.

J. Turian, B. Wellington, and I.D. Melamed. 2007. Scalable discriminative learning for natural language parsing and translation. *Advances in Neural Information Processing Systems*, 19:1409.

R. Zens, S. Hasan, and H. Ney. 2007. A systematic comparison of training criteria for statistical machine translation. In *Proc. of EMNLP*, pages 524–532.