Transductive Minimum Error Rate Training for Statistical Machine Translation

Yinggong Zhao[1]*, **Shujie Liu**[2], **Yangsheng Ji**[1], **Jiajun Chen**[1], **Guodong Zhou**[3]
[1]State Key Laboratory for Novel Software Technology at Nanjing University
Nanjing 210093, China
{zhaoyg,jiys,chenjj}@nlp.nju.edu.cn
[2]School of Computer Science and Technology, Harbin Institute of Technology
Harbin, China
shujieliu@mtlab.hit.edu.cn
[3]School of Computer Science and Technology, Soochow University
Suzhou, China
gdzhou@suda.edu.cn

Abstract

This paper investigates parameter adaptation in Statistical Machine Translation(SMT). To overcome the parameter bias-estimation problem with Minimum Error Rate Training(MERT), we extend it under a transductive learning framework, by iteratively re-estimating the parameters using both development and test data, in which the translation hypotheses of the test data are used as pseudo references. Furthermore, in order to overcome the over-training and unstableness problems respectively in employing such pseudo references, a termination criterion using a hyper-parameter and a Minimum Bayes Risk(MBR)-based hypothesis selection method are proposed in our work. Experimental results show that the transductive MERT method could yield significant performance improvements over a strong baseline on a large-scale Chinese-to-English translation task.

1 Introduction

Machine translation (MT) is the automatic translation from one natural language into another using computer, while SMT is an approach to MT that is characterized by the use of machine learning methods (Lopez, 2008). Nowadays, SMT is usually built on a log-linear model (Och and Ney, 2002), which can be abstracted into two steps: the first one is model training, i.e., learning features from large collection of bilingual parallel corpus; and the other is parameter estimation, in which the feature weight is tuned on an independent development dataset.

More specifically, for each source sentence f, we search for its final translation e^* among all possible translations based on the following equation:

$$P(e^*|f) = \arg\max_e Pr(e|f) \quad (1)$$

Under the log-linear model, the posterior probability $Pr(e|f)$ can be decomposed as:

$$Pr(e|f) = p_\lambda(e|f)$$

$$= \frac{\exp(\sum_{m=1}^{M}(\lambda_m \cdot h_m(e,f)))}{\sum_{e'}\exp(\sum_{m=1}^{M}(\lambda_m \cdot h_m(e',f)))} \quad (2)$$

where each $h_m(e,f)$ is a feature function and λ_m is the corresponding weight for m=1,...,M.

In SMT, there are three sections of data: the training data for feature estimation, the development data for weight tuning, and the test data for final evaluation. However, these three parts may belong to different domains, leading to distribution variations, which indicates that the features and weights learned from the data may be bias-estimated. As a result, the adaptation for features and their corresponding weights are both important issues in SMT.

In this article we focus on the latter issue, i.e., the model parameter adaptation. From the viewpoint of machine learning, development data is labeled data used for parameter learning, while test data is unlabeled and applied for evaluation. In

*Part of this work is done during the first author's internship at Microsoft Research Asia.

Proceedings of the 5th International Joint Conference on Natural Language Processing, pages 641–648,
Chiang Mai, Thailand, November 8 – 13, 2011.

the previous works of transductive learning(Liu et al., 2010; Chan and Ng, 2007), the unlabeled data can be used to improve the model training so as to tackle the bias-estimation problem. Under such framework, the weight learned on both development and test dataset, in which the test dataset is constructed using n-best translations as pseudo references, moves towards the test data with regularization of development data, which alleviates the overtraining in normal MERT and matches the test data better.

The remainder of this paper is structured as follows: Related works on model adaptation in SMT are presented in Section 2, and our transductive MERT is proposed in Section 3. Experimental results are shown in Section 4, followed by conclusions and future work in the last section.

2 Related Work

Model adaptation in SMT has attracted increasing attentions in recent years. As mentioned in the previous section, corresponding to the two steps in SMT pipeline, there are two directions for adaptations.

The first one is feature adaptation, which tries to build model (translation model & language model) that could fit the development or test dataset better. This direction includes data selection (Lü et al., 2007; Hildebrand et al., 2005) and data weighting (Foster and Kuhn, 2007; Matsoukas et al., 2009). However, efficiency is the main obstacle for these methods (esp. data selection approach) since model building is time consuming.

The second direction is model parameter adaptation, which includes the transductive MERT method we propose in this article. Nevertheless, little attention has been paid to this direction to date. Mohit et al (2009) tried to build a classifier to predict whether or not a phrase is difficult. The language model weight is then adapted for each phrase segment based on this difficulty. In Li et al (2010), a related subset of development dataset is extracted for given test dataset. The test dataset is then translated under weight learned on this subset. Besides, Sanchis-Trilles and Casacuberta (2010) propose Bayesian adaptation for weight optimization based on a small amount of labeled test data, which is not necessary in our work.

The most similar previous work with ours is Ueffing et al (2007), who also propose a transductive learning framework for SMT. However, our method is different from their in the following three aspects:

Firstly, our method focuses on model parameter adaptation, while Ueffing et al (2007) pays attention to feature adaptation. In their work, the training model is rebuilt by combining original training data with n-best translation outputs of development and test data, in order to overcome the data sparseness problem. In contrast, we try to solve the parameter bias-estimated problem using the information of both development and test data.

Secondly, the parameter adaptation problem is more complicated in SMT, since overtraining is serious due to the limited size of development data. In this work, we use hyper-parameter to indicate the overtraining in the estimation step.

Finally, our method is more efficient than adaptation on the translation & language model. In Ueffing et al.(2007), training model building is necessary for each round, which is time consuming. By comparison, the running time is much shorter for our method, since no model building is required, although it is still longer than simple one-pass translation under baseline.

3 Transductive MERT for Machine Translation

3.1 Minimum Error Rate Training

In SMT, given a development dataset containing source sentences F_1^S with corresponding reference translations R_1^S, the purpose of MERT (Och, 2003) is to find a set of parameters λ_1^M which optimizes an automated evaluation metric (e.g., BLEU) under a log-linear model:

$$\hat{\lambda}_1^M = \arg\min_{\lambda_1^M} \sum_{s=1}^{S} (Err(R_s, \hat{E}(F_s; \lambda_1^M))) \quad (3)$$

in which the number of errors in sentence E is obtained by comparing it with a reference sentence R using function $Err(R, E)$ and

$$\hat{E}(F_s; \lambda_1^M) = \arg\max_{E} \sum_{s=1}^{S} (\lambda_m h_m(E, F_s)) \quad (4)$$

As shown in Algorithm 1, the decoder translates development dataset under current weight(default weight for first round), and generates N-best translation hypotheses for each sentence. The weight is then updated according to equation 3. This procedure repeats until performance converges.

Algorithm 1 MERT for SMT

Input: Development data :$\{F_1^S, R_1^S, C_1^S\}$
Set λ= init-weight and C_1^S=$\{\}$
Translate F_1^S and get N-Best list L_1^S
while C_1^S != $C_1^S \bigcup L_1^S$ **do**
  $C_1^S = C_1^S \bigcup L_1^S$
  Update λ using translation candidates C_1^S
  Translate F_1^S using λ to generate N-Best list L_1^S
end while

3.2 Transductive MERT

The basic idea of transductive learning is to use predicted labels from unlabeled data to improve learning performance. Based on the this assumption and normal MERT method, our transductive MERT (T-MERT) works as follows: Firstly, the feature weight is estimated on the development data with references. Test dataset is then translated using current weight. For each source sentence of test data, its 4-best translations are used as pseudo references. The feature weight is further re-estimated based on both the development dataset and the test dataset with pseudo references. Meanwhile, the pseudo references of test dataset are replaced in each round, while the development dataset is fixed throughout the procedure. The whole process runs M rounds so that we could get M different results, which are used in the hypothesis selection step (discussed in section 3.4).

As shown in Algorithm 2, the T-MERT algorithm could be divided into two loops: in the outer loop (outer-translation step), the test dataset is translated under current weight and new pseudo labeled test dataset is constructed; while in inner loop (inner-MERT step), the parameter weight is learned from the combined dataset. Meanwhile, there still remains two problems in algorithm 2: when the loop will terminate in inner-MERT step, and how we can select final hypothesis from the multiple results for test data T. These two issues will be discussed respectively in following parts of this section.

3.3 Stop Criterion

In MERT, the loop terminates until the performance converges. While in T-MERT, the weight would be overtrained to the pseudo references of the test data, which could not guarantee that the translations obtained in each iteration are good e-

Algorithm 2 Transductive MERT for SMT

Input: Development data $\{D_1^S, R_1^S, C_1^S\}$, Test data $\{T_1^W\}$, total round M
Let L=$\{D_1^S, R_1^S, C_1^S\}$ and U= $\{T_1^W\}$
Do MERT based on L and get weight λ
Let C_1^S=$\{\}$
Translate U under λ and get N-Best list N_1^W
for $i = 1$ **to** M **do**
  Select 4-best translations to build $\tilde{U}_1^W$ from N_1^W
  Let $L = \{\{D_1^S, R_1^S, C_1^S\} \bigcup \{T_1^W, \tilde{U}_1^W, N_1^W\}\}$
  Set λ= init-weight
  repeat
    Translation L and get N-Best translations LB_1^{S+W}
    Let $L = L \bigcup LB_1^{S+W}$
    Update λ on L
  until Certain condition satisfies(Section 3.3)
  Translate U under λ and get N-Best list N_1^W, in which we select 1-best translation as T_i
end for
Select final translation(Section 3.4) from collections T_i (i=1, . . . , M)

nough. Here we introduce a hyper-parameter H to indicate the overtraining. In each inner round i, let SD_i stands for the BLEU score of development data D, SpT_i represents the BLEU score of test data T under pseudo references and $SDpT_i$ indicates the BLEU score of combined dataset L , then we define the hyper-parameter H_i as follows:

$$H_i = \frac{SD_i}{SD_{i-1}} \cdot exp\frac{SpT_i}{SpT_{i-1}} \cdot exp\frac{SDpT_i}{SDpT_{i-1}} \quad (5)$$

Here, H_i represents the relative improvement between the performance of inner round i and that of the previous round. A smaller H_i value indicates the inner-MERT turns to be converged on combined dataset L, showing that the weight would be overtrained. Due to the fact that the test dataset owns no references, we cannot attain its BLEU score in each round. As an alternative, we could only obtain SD_i, SpT_i and $SDpT_i$, as shown in above equation. In optimization step of normal MERT, what we need to do is to update the parameter to maximize the score on development data. While here we encounter the overtraining, we use ratio of scores to indicate the training. Instead of maximizing the score, we want to optimize the relative improvement of the system performance. In

T-MERT, we observe that the performance is always the best when the inner-MERT terminates as H_i reaches peak[1].

3.4 Hypothesis Selection with Minimum Bayes Risk(MBR)

From T-MERT algorithm, we can get M different results from M outer-translation rounds. Due to intrinsic property and the randomness in MERT, the results from outer-translation step of T-MERT are not quite stable, making the hypotheses selection a necessity.

According to (Ehling et al., 2007), for each source sentence with N different translations, we could select the final translation based on the following Minimum Bayes Risk principal:

$$\hat{e} = \arg\min_{e}\{\sum_{e'}(Pr(e'|f)\cdot(1-BLEU(e',e)))\} \quad (6)$$

Here $Pr(e'|f)$ denotes the posterior probability for translation e' and $BLEU(e',e)$ represents the sentence-level BLEU score for e' using e as reference.

However, since the translation hypotheses are generated under different groups of weights, the corresponding posterior probability is no longer comparable. Here we simplify this problem under the assumption that all available translations are generated equally. Then equation 6 could be converted into:

$$\hat{e} = \arg\min_{e}\{\sum_{e'}(1 - BLEU(e',e))\} \quad (7)$$

Based on equation 7, we can select the hypothesis from the collections of translations efficiently. And the primary purpose of using MBR selection in this work is to stabilize the translation performance, as we select final result using only sentence-level BLEU scores between different hypotheses.

4 Experimental Results

4.1 Experimental Setup

In the experiments, we re-implement a hierarchical phrase-based decoder based on (Chiang, 2005). The word alignment is trained by GIZA++ under an intersect-diag-grow heuristics refinement. The plain phrases are extracted from all bilingual training data available from LDC, including LDC2002E18, LDC2003E07, LDC2003E14, LDC2004E12, LDC2004T08, LDC2005E83, LDC2005T06, LDC2005T10, LDC2006E26, LDC2006E34, LDC2006E85, LDC2006E92, and LDC2007T09, which consists in total of about 8.5M sentence pairs while hierarchical rules are only extracted from selected data sets, including LDC2003E14, LDC2003E07, LDC2005T10, LDC2006E34, LDC2006E85, and LDC2006E92, which contain about 467K sentence pairs. We build the 5-gram language model on the English section of all bilingual training data together with the Xinhua portion of the English Gigaword corpus.

[1] And we find that in experiments, hyper-parameter H_i of the second round is always maximal.

Table 1: Statistics on development and test data sets.

DATA SET	#SENTENCE	#WORD
MT03	919	36,021
MT05	1,082	43,765
MT06	1,664	38,209
MT08	1,357	33,042

The development and test dataset pairs are selected from NIST2003 (MT03), NIST2005 (MT05), NIST2006 (MT06) and NIST2008 (MT08). The data statistics are shown in Table 1. In the experiments, all translation results are measured in case-insensitive BLEU scores (Papineni et al., 2002).

4.2 Results under Transductive MERT

Figure 1 and Figure 2 show the hyper-parameter $H(iter_{number})$ for each iteration in the 10-round inner-MERT step. In both figures, MT03 is development dataset while MT05 & MT08 are test datasets. We find that the hyper-parameter H always reaches the peak at the 2nd iteration, showing fast convergence during parameter estimation on the combination of development data and pseudo labeled test data. Similar phenomenon could also be observed on other dataset pairs.

Here, the reason might be that the pseudo translate references for the test data are generated with the current SMT model and its parameters. So the newly generated translation references on the test data are intuitively similar to translations obtained using the current model parameters. When we re-estimate the parameters on the combined dataset starting from the initial parameters, the learning procedure can quickly fit the newly generated data. While parameter estimation step continues iteratively, the learning algorithm may fa-

vor those incorrectly generated translation references, which makes the overtraining more serious and hurts the final performance. By applying the hyper-parameter as the stop metric, we could control the learning procedure to avoid the overtraining.

We can also review the roles that the development and test datasets play in the procedure of avoiding over-training. The reason for that we transductively generate translations as pseudo references for test data is that we expect the estimation procedure biases towards the test data when incorporated in the learning procedure. Meanwhile, the development data also plays an important role in the learning process. Because development data owns true references, it acts as a regularization term to ensure that the feature weight will not excessively biased toward the test data with generated pseudo references in the learning procedure.

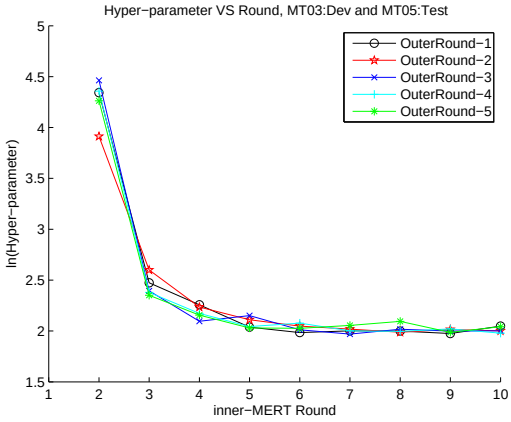

Figure 1: Hyper-parameter of 10 inner-MERT loops under 5 outer-translation rounds(OutRound-i) in T-MERT algorithm(without MBR), MT03:Dev and MT05:Test.

Based on the above discussion, we also compare results under different rounds for inner-MERT to verify the role of the hyper-parameter. As shown in figure 3(MT03 development and MT05 test) and figure 4(MT03 development and MT08 test), the results under T-MERT with 2-rounds inner-MERT are always best among different rounds, which is close to the baseline in figure 3 and much better in figure 4. Here the baseline for the test dataset is translated under weight learned from normal MERT on the development data, and remains constant for the following parts. For both

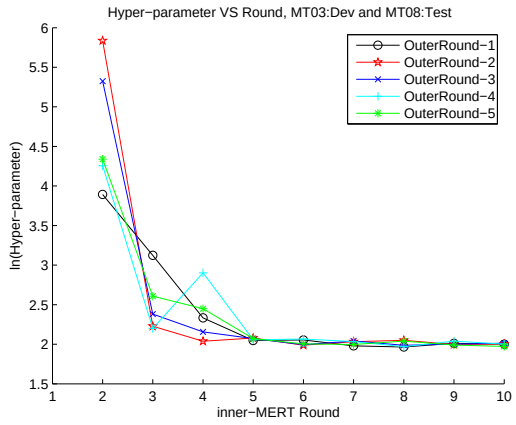

Figure 2: Hyper-parameter of 10 inner-MERT loops under 5 outer-translation rounds in T-MERT algorithm(without MBR), MT03:Dev and MT08:Test.

figures, we observe that 1-round inner-MERT is not sufficient to learn the weight well, while inner-MERT using more than 2 rounds leads to significant overtraining, which is consistent with the results obtained from the hyper-parameter.

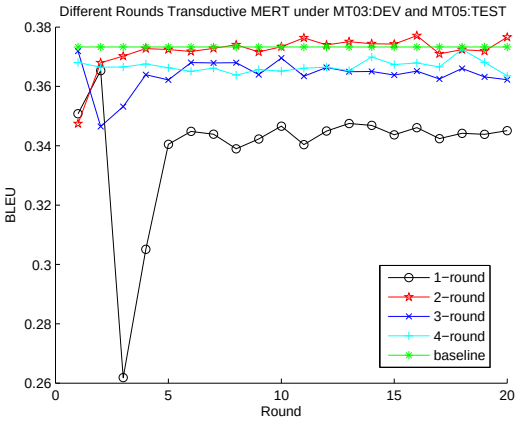

Figure 3: Results under T-MERT algorithm(without MBR) with fixed inner-MERT loops from 1 to 4, 20 outer-translation rounds, and baseline. MT03:Dev and MT05:Test.

Although the score of T-MERT under 2-round inner-MERT is comparable to or even better than baseline, the performance is still unstable, changing drastically for different rounds of outer-translation step (over 4 BLEU points for MT03:Dev and MT05:Test, and even larger for MT03:Dev and MT08:Test). We use the MBR s-

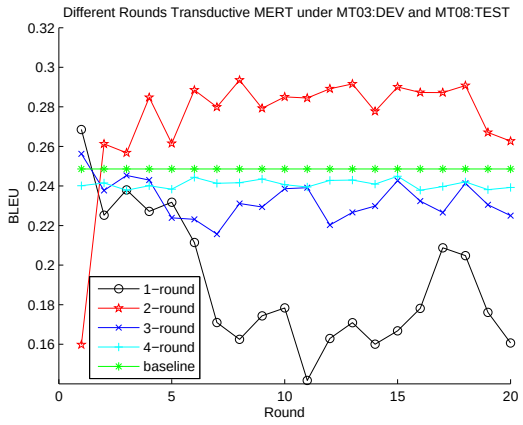

Figure 4: Results under T-MERT algorithm(without MBR) with fixed inner-MERT loops from 1 to 4, 20 outer-translation rounds, and baseline. MT03:Dev and MT08:Test.

election proposed in section 3.4 to choose a suitable hypothesis, and the corresponding results are shown in figure 5 and figure 6. It could be found that as the number of outer-translation rounds increases, the algorithm generates more groups of translation outputs, from which the performance under MBR selection turns to be more and more stable.

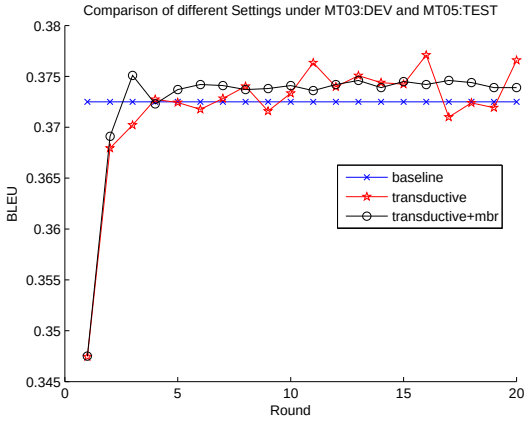

Figure 5: Result of baseline,T-MERT under 2 inner-MERT rounds without and with MBR selection. MT03:DEV and MT05:Test.

The above parts discuss two solutions for the problems we encounter in the transductive MERT, i.e., the inner-MERT stop criterion and MBR selection. We further evaluate our method (under 2 round inner-MERT and MBR selection) on all

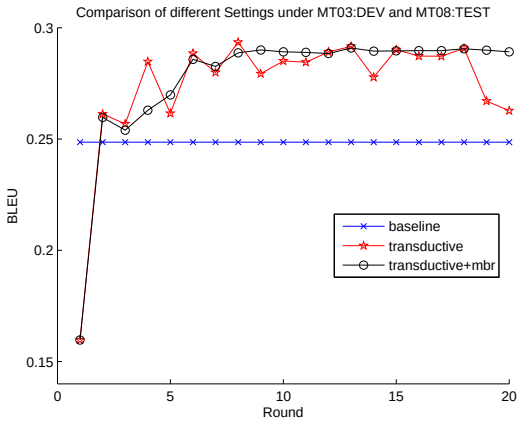

Figure 6: Result of baseline,T-MERT under 2 inner rounds without and with MBR selection. MT03:DEV and MT08:Test.

dataset pairs, which is any pair of MT03, MT05, MT06 and MT08. The final results are shown in table 2, from which we observe that for the dataset pair MT03 and MT05 the result under T-MERT is close to baseline, while for the pair MT03 and MT08, the improvement is significant in both directions. The result is similar with the observation on these evaluation datasets, i.e., MT03 and MT05 are under similar distribution, while MT03 and MT08 are quite different. Generally speaking, MT03 and MT05 are both composed of only news data, while MT06 and MT08 are consisted of news and web-blog data. As we know, the news data is significantly different from the web-blog data. We can find that our method could achieve significant improvement on 9 of total 12 dataset pairs, indicating that the distribution variation between dataset pairs is quite a common phenomenon. For datasets under similar distribution, the baseline performance is close to oracle, which means that potential space for improvement under the adaptation is limited; while for datasets that are quite different, there is much room for further increase in performance, since the baseline weight estimated on development is seriously biased for the test.

Besides, we try one extra comparison, i.e., using same dataset for both development and test under T-MERT. The result is also shown in table 2. We find that for MT03 and MT05, the result under T-MERT is close to baseline (a little higher), while for MT06 and MT08, the result is fairly lower. The reason that the adapted result on MT03 is slightly higher than baseline is that the result in

Table 2: Results of baseline and T-MERT under MBR Selections for different dataset pairs. Here symbol ↑ shows that the improvement is significant, ↓ indicates decrease is significant, and | means no significant changes

DEV	MT03		MT05		MT06		MT08	
TEST	BASELINE	T-MERT	BASELINE	T-MERT	BASELINE	T-MERT	BASELINE	T-MERT
MT03	0.3914	0.3933(|)	0.3861	0.3908(|)	0.3731	0.3830(↑)	0.3586	0.3704(↑)
MT05	0.3733	0.3739(|)	0.3687	0.3724(|)	0.3592	0.3700(↑)	0.3414	0.3576(↑)
MT06	0.3358	0.3582(↑)	0.3344	0.3569(↑)	0.3636	0.3579(↓)	0.3504	0.3653(↑)
MT08	0.2486	0.2892(↑)	0.2543	0.2755(↑)	0.2774	0.2768(|)	0.2929	0.2809(↓)

each round is close to baseline, making it possible for the selection performance to be slightly higher than baseline, while for others the result in each round is lower than baseline. However, we do not hope that our method could be significantly better than baseline in this case, as baseline performance is also the oracle performance for the test dataset. We assume that we know the development and the test datasets are distinct in advance before applying the T-MERT algorithm.

5 Conclusion and Future Work

In this paper, we investigate the parameter adaptation issue in SMT. In particular, a transductive MERT algorithm is proposed to better explore both development and test datasets. Besides, a hyper-parameter is proposed to control the over-training problem in the parameter estimation step and a Minimum Bayes Risk (MBR)-based hypothesis selection method is adopted to stabilize the final performance. Compared with a state-of-the-art baseline, our method achieves significant and sustainable improvement.

In future, we plan to incorporate better hypothesis selection algorithms to choose high quality sentences from the test dataset, since sentences with bad translations would bring side effect during the learning procedure. Besides, we plan to further investigate the mechanism of transductive MERT in boosting the performance of SMT.

Acknowledgments

We thank Ning Xi and Shujian Huang for their meaningful suggestions. We would also like to thank the anonymous reviewers for their helpful comments. This work is supported by the National Natural Science Foundation of China (No.61003112) and the National Fundamental Research Program of China (2010CB327903)

References

Yee S. Chan and Hwee T. Ng. 2007. Domain adaptation with active learning forword sense disambiguation. In *In Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pp. 49–56, Prague, Czech Republic, 2007.

David Chiang. 2005. A hierarchical phrase-based model for statistical machine translation. In *In Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, pp. 263–270, Ann Arbor, 2005.

Nicola Ehling, Richard Zens and Hermann Ney. 2007. Minimum bayes risk decoding for bleu. In *In Proceedings of the ACL 2007 Demo and Poster Sessions*, pp. 101–104, Prague, 2007.

George Foster and Roland Kuhn. 2007. Mixture-model adaptation for smt. In *In Proceedings of the Second ACL Workshop on Statistical Machine Translation*, Prague, Czech Republic, 2007.

Almut Silja Hildebrand, Matthias Eck, Stephan Vogel and Alex Waibel. 2005. Adaptation of the translation model for statistical machine translation based on information retrieval. In *In Proceedings of EAMT*, Budapest, Hungary, 2005.

Philipp Koehn. 2004. Statistical significance tests for machine translation evaluation. In *In Proc. of the Conference on Empirical Methods in Natural Language Processing*, pp. 160–167, 2004.

Mu Li, Yinggong Zhao, Dongdong Zhang and Ming Zhou. 2010. Adaptive development data selection for log-linear model in statistical machine translation. In *In Proceedings of the 23rd International Conference on Computational Linguistics (Coling 2010)*, pp. 662–670, Beijing, 2010.

Yang Liu, Steve Hanneke and Jaime Carbonell. 2010. A theory of transfer learning with applications to active learning. Technical report, Machine Learning Department, Carnegie Mellon University, New Brunswick, MA, 2010.

Adam Lopez. 2008. Statistical machine translation. *ACM Computing Surveys*, 40(3), 2008.

Yajuan Lü, Jin Huang, and Qun Liu. 2007. Improving statistical machine translation performance by

training data selection and optimization. In *In Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing*, pp. 343–350, Czech Republic, 2007.

Spyros Matsoukas and Antti-Veikko I. Rosti and Bing Zhang. 2009. Discriminative corpus weight estimation for machine translation. In *In Proc. of the Conference on Empirical Methods in Natural Language Processing*, pp. 160–167, 2009.

Behrang Mohit, Frank Liberato and Rebecca Hwa. 2009. Language model adaptation for difficult to translate phrases. In *In Proceedings of the 13th Annual Conference of the EAMT*, pp. 160–167, 2009.

Franz Och. 2003. Minimum error rate training in statistical machine translation. In *In Proceedings of the 41th Annual Meeting of the Association for Computational Linguistics (ACL)*, Sapporo, Japan, 2003.

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

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

German Sanchis-Trilles and Francisco Casacuberta. 2010. Log-linear weight optimisation via bayesian adaptation in statistical machine translation. In *In Proceedings of the 23rd International Conference on Computational Linguistics (Coling 2010)*, pp. 662–670, Beijing, 2010.

Nicola Ueffing, Gholamreza Haffari and Anoop Sarkar. 2007. Transductive learning for statistical machine translation. In *In Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pp. 25–32, Czech Republic, 2007.