Language Model Weight Adaptation Based on Cross-entropy for Statistical Machine Translation

Yinggong Zhao, Yangsheng Ji, Ning Xi, Shujian Huang and Jiajun Chen

State Key Laboratory for Novel Software Technology at Nanjing University, Nanjing 210093, P.R.China
{zhaoyg, jiys, xin, huangsj, chenjj}@nlp.nju.edu.cn

Abstract. In this paper, we investigate the language model (LM) adaptation issue for Statistical Machine Translation (SMT). In order to overcome the weight bias on the LM obtained from the development data, a simple but effective method is proposed to adapt the LM for diverse test datasets by employing the cross entropy of translation hypotheses as a metric to measure the similarity between different datasets. Experimental results show that the cross entropy of a test dataset is closely correlated with the bias in estimating the language models and our adaptation strategy significantly outperforms a strong baseline.

Keywords: Statistical machine translation, Language model, Weight adaptation, Cross-entropy

1 Introduction

Language modeling is applied in many natural language processing (NLP) applications, including automatic speech recognition (ASR) and SMT. In reality, we often encounter the scenario in which the performance of language model learned from given dataset changes drastically among different datasets. Many adaptation techniques have been proposed to tackle this problem in the field of ASR. A similar situation arises with respect to SMT. In SMT we build language model from large amounts of monolingual data but incorporate it in the translation task of the dataset that is not well covered by the model. This inconsistency inevitably affects the SMT training procedure, making adaptation techniques a necessity.

Different from other tasks, language model is incorporated under a log-linear framework in SMT. Specifically, for each source sentence f, we search for the final translation e^* among all possible candidates under the following equation:

$$P(e^*|f) = \arg\max_e Pr(e|f) \tag{1}$$

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

$$\begin{aligned} Pr(e|f) &= p_\lambda(e|f) \\ &= \frac{\exp(\sum_{m=1}^{M}(\lambda_m \cdot h_m(e,f)))}{\sum_{e'}\exp(\sum_{m=1}^{M}(\lambda_m \cdot h_m(e',f)))} \end{aligned} \tag{2}$$

where $h_m(e,f)$ is a feature function and λ_m is related weight for $m = 1,\ldots,M$.

Under the above framework, we tune the model weight on an independent development dataset, and then we use the obtained weight to translate diverse datasets whose domain or related information might be previously unknown. It is noticeable that the weight obtained from Minimum Error Rate Training (MERT) matches the development dataset well, whereas it would be bias-estimated

for others. Although the value of each feature's weight represents its importance in the decoding procedure, such type of importance might vary for different datasets under a specific language model. In this article we concentrate on the bias-estimation of language model weight, i.e., the difference between the oracle and actual LM weight as shown in section 3. We measure the similarity between datasets based on cross-entropy of translation output according to a given language model, adapt the LM weight based on the ratio of the cross-entropy and obtain the final results through a second-pass translation. Our LM weight adaptation method is also related with density ratio estimation, as mentioned in (Tsuboi et al., 2008), in which reweighting approach is proposed to overcome the bias due to the different distribution of test and training data.

The remainder of this paper is organized as follows: Related work of LM adaptation is presented in Section 2. In Section 3 we discuss the problem of LM weight bias-estimation in machine translation. And in Section 4, cross-entropy is proposed as a metric for measuring the similarity between different datasets and we further present our adaptation method. Experimental results are shown in Section 5. We conclude and present several directions for future work in the last section.

2 Related Work

Nowadays LM adaptation in SMT has been paid lots of attentions. There are two main categories for this problem.

The first one is data selection, i.e., when given a test dataset and a large general corpus, which tries to extract sentences from the whole corpus that are relevant to the test dataset under some metric. There are two main approaches for the measurement: One is to apply tf-idf metric (Hildebrand et al., 2005; Lü et al., 2007; Zhao et al., 2004), which arises from information retrieval; while for the other approach cross-entropy (perplexity) is adopted for selection, as reported in (Axelrod et al., 2011; Moore and Lewis, 2010).

The second is model weighting. The main idea is to assign appropriate weight to each model according to the similarity between the model corpus and test dataset. In this approach, the models could be built from domain-specific corpus (Koehn and Schroeder, 2007) when domain of the test dataset is known, or from datasets that belong to different sources (Foster and Kuhn, 2007; Lü et al., 2007) when it is unavailable in advance. Such weighting method even could be apply to either each sentence from the training corpus (Matsoukas et al., 2009) or phrase pair from the phrase-table (Foster et al., 2010). Besides, the work of (Mohit et al., 2009; Mohit et al., 2010) also belongs to such scenario, in which they attempt to build a classifier to predict whether or not a phrase is difficult, then the LM weight is updated for each phrase segment according to its difficulty.

The methods mentioned above try to overcome the difference between the training and the test data. However, the bias between the development and the test data is also an open issue. Not much attention has been paid to the such weight adaptation. In Li et al. (2010), the model weight is tuned on a subset of the development set, which is extracted based on the relevance to the test set.

In this paper, different from (Li et al., 2010), we focus on the adaptation of LM weight only, as LM is one of the key components of SMT and has its own characteristic. In our work, we adopt cross-entropy as a metric, just as (Axelrod et al., 2011; Moore and Lewis, 2010), to measure the similarity between different datasets. However, only LM weight is adjusted during the adaptation, and no extra model needs to be built. Although our method is quite simple and straightforward, the improvements obtained from the adaptation show that the bias-estimation of LM weight due to the difference between development and test dataset is also quite an important issue in SMT.

3 Language Model Weight Mismatch in Statistical Machine Translation

As model weight is tuned on development dataset only but applied to various test datasets, the mismatch between development and test is inevitable. To verify the LM weight bias-estimation

for different dataset pairs, we conduct the following experiment in this section: for development dataset pair D(development) and T(test), we firstly learn the weight via MERT on D. Then with all other feature weights fixed, we translate T and record the change of BLEU score compared with baseline during the step-by-step modification of the LM weight by starting from the initial weight with a constant value each time.

Based on the above approach, we use four dataset pairs for comparison under a large scale experiment setting (Section 5.1). Figure 1 shows the relation between the BLEU score of the test datasets and the corresponding LM weight. Specifically speaking, each point (x, y) in the figure means that under new LM weight $x * baseline\text{-}LM\text{-}weight$, the BLEU score of test dataset under new weight changes y points compared with baseline. We could observe that for some datasets pair like MT03 as development and MT08 as test, the weight is seriously bias-estimated. The detailed comparison is shown in table 1, in which the oracle performance represents the maximal BLEU score obtained when we manually modify the LM weight. The significant difference between baseline and oracle result (about 3 BLEU points) shows much room for potential improvement. Meanwhile, the weight fits well for dataset pair MT03(development) and MT04(test), since the baseline performance is close to the oracle.

Based on the above observation, we find that the LM weight mismatch is a common phenomenon in SMT. And the bias-estimation is different for various dataset pairs. Thus it is necessary to propose a metric that could measure the similarity between datasets and an adaptation strategy on LM weight, as we will discuss in the next section.

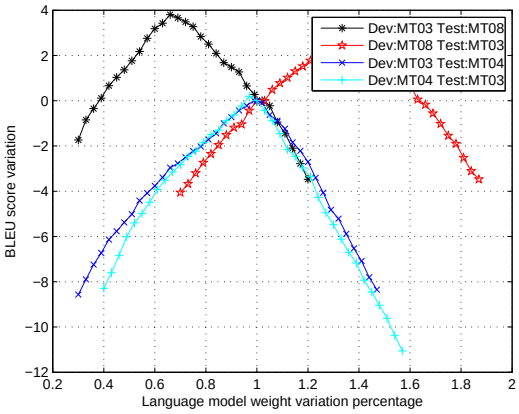

Figure 1: The variation of BLEU score (in value) vs. variation of language model weight (in percentage)

DEV	TEST	Baseline	Oracle
MT03	MT04	37.54	37.55(+0.01)
MT04	MT03	38.76	38.96(+0.20)
MT03	MT08	24.86	28.66(+3.80)
MT08	MT03	35.86	38.77(+2.91)

Table 1: Comparison between baseline and oracle performance under language model weight for different dataset pairs.

4 Dynamic Language Model Weight Adaptation Under Cross Entropy

Entropy is used as a metric to show how much information one dataset contains. Given sentence $X_i = (x_1, x_2, \ldots, x_n)$, the corresponding cross-entropy under specific language model lm could

be calculated as:

$$H(X_i) = -\frac{1}{n} \log P_{lm}(x_1, x_2, \ldots, x_n) \tag{3}$$

Given two datasets and one language model, we could use cross-entropy to identify which dataset matches the language model better. As the language model is built on target language in SMT task, we can use the entropy of the translation outputs for each dataset as a measurement of the dataset. Given a language model and two datasets (development and test), the model weights are tuned through MERT on development dataset. Then we compute their cross-entropy after translating both datasets under current weight. Specifically speaking, for a dataset X that contains multiple sentences, we obtain its cross-entropy according to the following equation:

$$H(X) = -\frac{\sum_i \sum_j \log P_{lm}(X_i^j)}{\sum_i \sum_j length(X_i^j)} \tag{4}$$

in which, X_i^j denotes the j_{th} best translation or reference for the i_{th} sentence in the dataset. As the decoder generates translation outputs together with corresponding feature vectors, $\log P_{lm}(X_i^j)$ could be viewed as equivalent to the language model feature.

According to the property of cross-entropy, we can know how the dataset fits the language model. Empirically speaking, a small cross-entropy value indicates a well-matching between the language model and dataset. The language model could thus play a more important role in the translating procedure, which further reveals a large value relatively. Hence, if the test data matches language model better than the development data, the language model weight might be under-estimated; otherwise it would be over-estimated. So we could conclude that cross-entropy difference can be used as a metric for how well the LM weight is estimated.

However, we encounter two problems: Firstly, how can we estimate the degree to which the LM weight is bias-estimated and the second is how we can adjust the weight appropriately. Here we hold the straightforward opinion that the difference of the cross-entropy between test and development data can be a metric for the LM bias-estimation. For the adaptation on the language model weight, we propose an effective method that merely uses cross-entropy. Let D be the development dataset and T be the test dataset. The adaptation approach is shown as follows:

1. Train a log-linear model based on D and obtain feature weight W.
2. Translate D using W and calculate the cross-entropy of D as $H(D)$, similarly translate T and obtain $H(T)$.
3. Modify the LM weight in W_{lm} by:

$$W_{lm} = W_{lm} \frac{H(D)}{H(T)}$$

   and get new weight W'.
4. Translate T again under W' and get the final result.

In the third step, we use the ratio of the entropy of the development and the test dataset for weight adjustment, as it could reflect the variance between these two datasets. It is known that in development dataset each sentence owns references, the reason we use entropy of translation outputs rather than references for development is that in real application we usually translate datasets without references, although it is included for standard SMT evaluation datasets. In fact we could observe that the adaptation result based on the cross-entropy of translation outputs is consistent with that based on cross-entropy of the references, as shown in section 5.2.

5 Experiments

5.1 Experiment Settings

We implement a hierarchical phrase-based decoder according to Chiang (2005). The development data includes NIST 2003 (MT03), NIST 2004 (MT04), NIST 2005 (MT05), NIST 2006 (MT06) and NIST 2008 (MT08). Besides the above four datasets, the test datasets contain all portions of MT06, including newswire (MT06nw), newsgroup (MT06wg) and weblog (MT06wl), and two portions of MT08, including newswire (MT08nw) and webgroup (MT08wg). The statistics are shown in Table 2. All results are measured in case-insensitive BLEU4 (Papineni et al., 2002).

Dataset	MT03	MT04	MT05	MT06	MT08	MT06bc	MT06nw	MT06ng	MT08nw	MT08wg
#Sentence	919	1,788	1,082	1,664	1,357	565	616	483	691	666
#Word	24,900	50,061	30,512	38,984	33,259	11,884	17,971	9,146	18,124	15,145

Table 2: Statistics on development and test datasets.

In the experiments, the training corpus includes LDC2002E18, LDC2003E07, LDC2003E14, LDC2004E12, LDC2004T08, LDC2005E83, LDC2005T06, LDC2005T10, LDC2006E26, LDC2006E34, LDC2006E85, LDC2006E92, and LDC2007T09, which consists of about 8.5M sentence pairs. The word alignment result is trained by GIZA++ in both directions and refined under intersect-diag-grow heuristics. The plain phrases are extracted from the all bilingual training data, while hierarchical rules are only extracted from selected datasets, including LDC2003E14, LDC2003E07, LDC2005T10, LDC2006E34, LDC2006E85, and LDC2006E92, which covers nearly 467K sentence pairs. We further train the 5-gram language model over the English part of training data plus Xinhua portion of the English Gigaword corpus.

5.2 Adaptation on 1-best Translation Result

In this part, we will evaluate the performance of our method introduced in section 4. Under each development dataset, we calculate the cross-entropy of all test datasets, which are displayed in table 3. The results of both baseline and under our adapted method are also presented in table 5.

DEV	MT03		MT04		MT05		MT06		MT08	
TEST	Baseline	Adapted	Baseline	Adapted	Baseline	Adapted	Baseline	Adapted	Baseline	Adapted
MT03	1.8842	1.8842	1.8507	1.8701	1.9953	1.9958	1.8058	1.7992	1.8800	1.8564
MT04	1.7556	1.7353	1.7264	1.7264	1.8720	1.8482	1.6900	1.6679	1.7600	1.7191
MT05	1.8880	1.8884	1.8621	1.8776	2.0022	2.0022	1.8109	1.8091	1.8759	1.8513
MT06	1.9287	1.9361	1.8997	1.9185	2.0459	2.0571	1.8408	1.8408	1.9224	1.9012
MT08	2.1462	2.1787	2.1176	2.1550	2.2890	2.3488	2.0376	2.0587	2.1224	2.1224
MT06bc	1.8324	1.8260	1.8262	1.8150	1.9425	1.9292	1.7657	1.7536	1.8352	1.7995
MT06nw	1.8480	1.8412	1.8175	1.8294	1.9535	1.9469	1.7607	1.7502	1.8378	1.8108
MT06ng	2.2962	2.3440	2.2660	2.3069	2.4436	2.5170	2.1692	2.2047	2.2637	2.2737
MT08nw	2.0602	2.0786	2.0208	2.0556	2.1913	2.2304	1.9548	1.9665	2.0271	2.0166
MT08wg	2.2649	2.3191	2.2544	2.2995	2.4204	2.5120	2.1511	2.1815	2.2527	2.2664

Table 3: .Cross entropy of test datasets on different development datasets, under large scale setting.

From table 5, we may find that the cross-entropy is quite close for some dataset pairs like MT03 and MT05 , which indicates that the adapted score would change little compared with baseline. While for the pair like MT03 and MT08, the remarkable difference means that we can achieve significant improvement (1.60 BLEU points for MT08 test and MT03 development, and 0.99 BLEU points for the reverse). We also obtain similar results on the other dataset pairs, including all separate portions of MT06 and MT08 whose genre information is available. Table 6 displays the oracle test performance in each dataset pair. We can observe that oracle performance for MT05(test) under MT03(development) is 37.54, while the baseline is 37.33, which is consistent with the ratio of cross-entropy between two datasets.

DEV:TEST	Method	1-gram	2-gram	3-gram	4-gram	BP	BLEU	TER
MT03:MT08	Baseline	0.7866	0.4155	0.2276	0.1278	-0.2282	0.2486	0.5904
MT03:MT08	Adapted	0.7708	0.4033	0.2186	0.1222	-0.1318	0.2646	0.5910
MT08:MT03	Baseline	0.7703	0.4611	0.2773	0.1679	0.0000	0.3586	0.5912
MT08:MT03	Adapted	0.7851	0.4733	0.2860	0.1735	0.0000	0.3685	0.5682

Table 4: Detailed analysis of BLEU scores, including n-gram precision and length penalty and TER scores, based on dataset pair of MT03 and MT08.

DEV	MT03		MT04		MT05		MT06		MT08	
TEST	Baseline	Adapted	Baseline	Adapted	Baseline	Adapted	Baseline	Adapted	Baseline	Adapted
MT03	39.14	39.14 (|)	38.77	38.45 (↓)	38.61	38.69 (|)	37.31	37.44 (|)	35.86	36.85 (↑)
MT04	37.52	36.74 (↓)	37.93	37.93 (|)	36.72	36.12 (↓)	35.81	36.84 (↑)	34.66	36.23(↑)
MT05	37.33	37.37 (|)	36.94	37.24(↑)	36.87	36.87 (↑)	35.93	36.07(|)	34.15	35.29 (↑)
MT06	33.58	34.04 (↑)	33.63	35.13 (↑)	33.44	33.49 (|)	36.36	36.36 (↑)	35.04	35.87 (↑)
MT08	24.86	26.46 (↑)	24.18	27.03 (↑)	25.43	26.65 (↑)	27.74	28.86 (↑)	29.29	29.29 (|)
MT06bc	24.22	27.20 (↑)	23.77	27.70 (↑)	24.64	26.26 (↑)	27.37	28.14 (↑)	28.87	28.43 (↓)
MT06nw	40.36	39.91 (↓)	39.97	40.74 (↑)	39.85	39.72 (|)	39.57	40.26 (↑)	37.71	39.72 (↑)
MT06ng	33.81	33.65 (|)	34.23	34.71 (↑)	33.79	33.45 (↓)	36.72	36.53 (|)	35.58	36.61 (↑)
MT08nw	29.40	30.66 (↑)	28.95	31.32 (↑)	29.67	30.31 (↑)	32.47	33.31 (↑)	33.03	33.23 (↑)
MT08wg	18.78	21.09 (↑)	17.81	21.04 (↑)	19.74	20.91 (↑)	21.35	23.06 (↑)	22.72	23.13 (↑)

Table 5: Comparison between baseline and LM weight adaption method using 1-best translation on different dataset pairs, under large scale setting. Symbol(↑) indicates improvement over 0.2 BLEU points, (↑) indicates improvement over 0.2 BLEU points, (↓) means decline over 0.2 BLEU points, (|) shows no noticeable change.

We also calculate the entropy on translations after adaptation of all dataset pairs, which is also listed in table 3. From the results in table 3, we find that the cross entropy usually changes according to the ratio of cross-entropy of development and test datasets. Specifically, the cross entropy of test dataset increases as LM weight decreases, as shown in figure 2, in which we use the same dataset pairs as in section 3. The reason for the phenomenon in figure 2 is that when the LM weight increases, the language model turns to play a more important role in the whole SMT system. As a result, the decoder prefers to select the translations with higher LM scores, which are also with shorter length and smaller cross-entropy.

Furthermore, we want to know what the improvements could be under our adaptation method. Here we take the pair MT03 and MT08 as example, the details of the results are shown in table 4. We may observe that for the pair of MT03 as development and MT08 as test, the length penalty is quite large. Meanwhile our adaptation method could notably reduce such penalty and get significant improvement based on BLEU metric. Although the n-gram precision decrease in some sense,

	MT03	MT04	MT05	MT06	MT08
MT03	39.14	38.92	38.77	38.24	38.44
MT04	37.56	37.93	36.78	37.53	37.48
MT05	37.54	37.55	36.87	36.99	37.21
MT06	36.30	36.42	35.03	36.36	36.43
MT08	28.58	28.55	27.50	29.04	29.29
MT06bc	40.98	41.41	40.18	40.29	40.60
MT06nw	36.74	36.75	35.40	36.74	36.60
MT06ng	28.15	28.31	27.22	28.77	28.77
MT08nw	32.82	33.03	31.79	33.71	33.22
MT08wg	22.83	22.79	21.26	23.24	23.22

Table 6: Oracle performance of different dataset pairs under large scale setting.

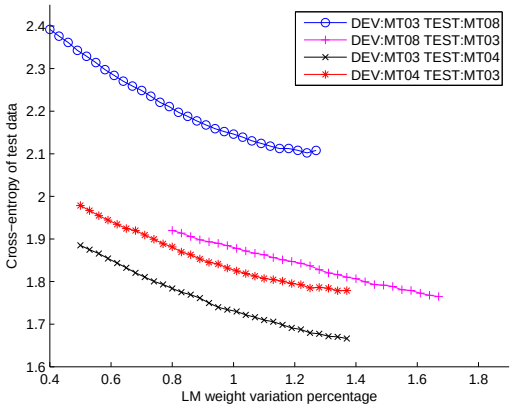

Figure 2: The cross entropy of test vs. LM weight variation in percentage for different dataset pairs

the gain from the length penalty decrease could counteract reduction on the precision. While for the case in which MT08 as development and MT03 as test, the length penalty of both baseline and adapted results are equal, while n-gram precision of adapted method is higher than that of baseline, which leads to improvements on final performance. Meanwhile, we also apply another SMT metric TER (Snover et al., 2006) to evaluate the results of the dataset pair MT03 and MT08, as shown in table 4. When we use MT03 as development and MT08 as test, the TER result shows no improvement. This is consistent with observation from above discussion, as improvement for BLEU mainly comes from length penalty, not n-gram precision. Meanwhile, when we use MT08 as development and MT03 as test, we achieve significant improvement on the TER score. This inconsistency shows some potential difference between the TER metric and the BLEU metric.

However, for some dataset pairs, the adapted result is not so good as the baseline. The reason might be that the closeness of test and development measured through cross-entropy is more significant than the real difference. Taking MT03(development) and MT04(test) for example, from figure 1 we could find that the baseline is almost the same as oracle (0.01 BLEU points difference), while the ratio of the cross-entropy from table 3 is larger than our intuition, making the LM weight over-adapted and the performance decreased. Nevertheless, the results in table 5 show that our method works well for most dataset pairs (33 of 50 groups increase, while only 6 of 50 groups decrease). Although our adaption method is in a sense empirical, we believe it reflects the inherent relations in the LM adaptation.

Furthermore, we want to know the influence of the cross-entropy variation on BLEU score improvements. In figure 3, the X-axis represents the absolute value of relative change between development and test dataset (i.e., $|\frac{H(D)}{H(T)} - 1|$), and the Y-axis displays the improvements of the BLEU score under adaptation. We would observe that all five groups of points are well linear, showing strong correlations between adaptation improvements and cross-entropy difference. Based on above results, we can draw the conclusion that even if cross-entropy may not be the only factor that determines the bias-estimation of LM weight, it is still one of the most important.

5.3 1-best VS. N-best Translation Result Adaptation

In the above part, we utilize mere 1-best translation results for entropy calculation. We wonder what the result would be if more outputs are used. With MT03 as development, MT05 and MT08 as test respectively, we run adaptation under number from 1 to 20 best translations. Results in figure 4 show that the number of translation outputs shows little impact on the adaptation results, since the deviation between maximal and minimal score is quite small(less than 0.2 BLEU score points). And in the following parts, we adopt 1-best translation as default setting.

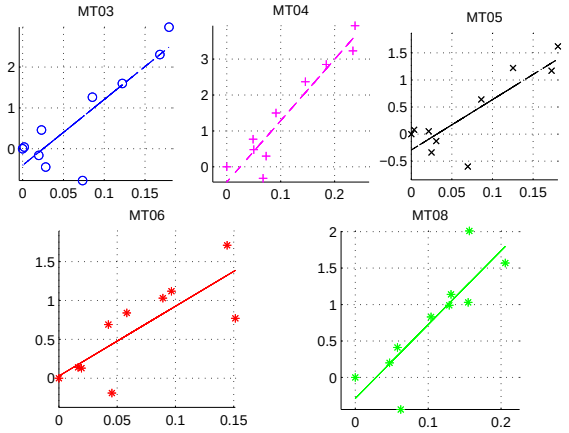

Figure 3: The variation of BLEU score (in value) vs. cross-entropy ratio between development and test dataset under each development datasets

DEV	MT03		MT04		MT05		MT06		MT08	
TEST	Baseline	Adapted	Baseline	Adapted	Baseline	Adapted	Baseline	Adapted	Baseline	Adapted
MT03	39.14	39.14 (|)	38.76	38.82 (|)	38.59	38.49 (|)	37.32	37.47 (|)	35.87	36.70 (↑)
MT04	37.52	37.04 (↓)	37.93	37.93 (|)	36.72	36.50 (↓)	35.80	36.53 (↑)	34.66	35.73 (↑)
MT05	37.33	37.11 (↓)	36.92	37.18 (↑)	36.87	36.87 (|)	35.90	36.22 (↑)	34.15	35.36 (↑)
MT06	33.58	33.85(↑)	33.62	34.51 (↑)	33.41	33.55 (|)	36.36	36.36 (|)	35.04	35.77 (↑)
MT08	24.86	26.23(↑)	24.18	26.30 (↑)	25.43	26.32 (↑)	27.75	28.93 (↑)	29.29	29.29 (|)
MT06bc	24.22	26.06 (↑)	23.77	26.44(↑)	24.64	26.18 (↑)	27.40	28.61 (↑)	28.87	28.48 (↓)
MT06nw	40.36	39.48(↓)	39.97	39.46 (↓)	39.78	39.63 (|)	39.57	40.54 (↑)	37.71	39.75 (↑)
MT06ng	33.81	34.08(↑)	34.23	34.72 (↑)	33.76	33.80(|)	36.73	36.74 (|)	35.59	36.08 (↑)
MT08nw	29.40	30.64 (↑)	28.95	30.85(↑)	29.67	30.32(↑)	32.49	33.10 (↑)	33.05	33.20 (|)
MT08wg	18.78	20.35 (↑)	17.81	20.03(↑)	19.74	20.88(↑)	21.35	22.65 (↑)	22.73	22.99 (↑)

Table 7: Comparison between baseline and LM weight adaption method using references on different dataset pairs, under large scale setting.

5.4 Adaptation on Translation References

In practice, entropy of translation outputs, rather than references, is used for adaptation. Nevertheless, we want to know whether there exists some difference between these two approaches. Table 7 shows the results under adaptation on entropy of references, while the related entropy is shown in Table 8. We could find that adapted results and the cross-entropy are both consistent with those of using 1-best translation, as in SMT the model weight is tuned in the way that tries to make translation outputs as close as possible to the references.

Dataset	MT03	MT04	MT05	MT06	MT08	MT06bc	MT06nw	MT06ng	MT08nw	MT08wg
Entropy	2.3450	2.2456	2.3019	2.3854	2.5778	2.2105	2.3690	2.6281	2.4955	2.6791

Table 8: Cross entropy of each dataset calculated on references.

5.5 Adaptation on Random Test Data

In our experiment, we always use the standard NIST datasets to evaluate the adaptation method. We also want to validate our method under more datasets in this section. Using MT03 as development, we build six extra test datasets by randomly selecting 50, 100, 300, 600, 1200 and 2000 sentences respectively from the collections of MT04, MT05, MT06 and MT08. Related results are shown in Table 9, in which improvements still could be achieved but not so significant as the results in table 5. Based on experimental results, we know that some datasets like MT04 and MT05

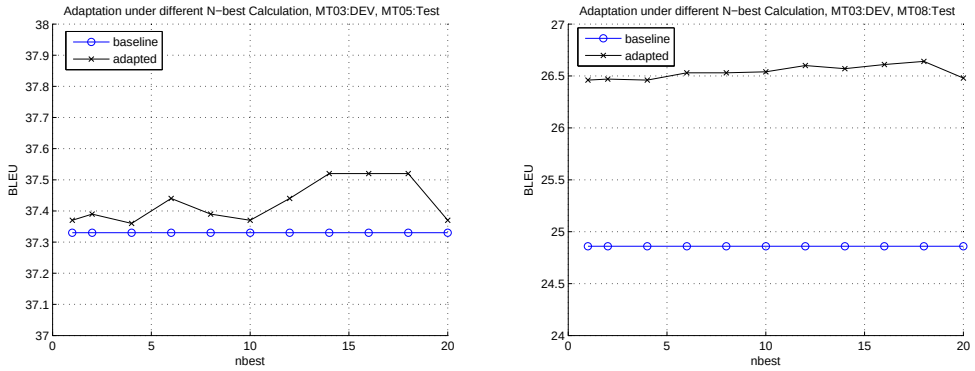

Figure 4: The adaptation results under entropy calculation on different number of translation outputs, with MT03 as Development and MT05 as Test(Left), MT08 as Test(Right)

Random	Baseline	Adapted	Oracle
50	34.52	35.03(+0.51)	35.05
100	34.79	34.33(-0.46)	34.94
300	33.06	33.38(+0.32)	33.82
600	33.48	33.72(+0.24)	34.91
1200	33.64	33.92(+0.28)	35.03
2000	33.72	34.04(+0.32)	35.22

Table 9: Results with MT03 as development and random selected datasets as test, under large scale setting

are close to the development MT03, while some others are different. One basic assumption for our adaptation method is that the dataset is composed of several documents, each belongs to a specific domain. Hence for the random datasets, their distribution is a mixture of multiple sources, making the adaptation performance not so significant as that on normal MT evaluation datasets.

6 Conclusion

In this article, we address the problem of LM weight mismatch between tuning and testing. In particular, cross-entropy on n-best translation hypotheses is adopted as a metric to indicate the bias-estimation in language modeling. Furthermore, an adaptation approach is proposed to adjust the LM weight using the ratio of cross-entropy between different datasets. Experimental results show that our cross-entropy based adaptation strategy significantly alleviates the bias problem of language modeling and significant improvements could be achieved when the test data is quite different from the development.

In this paper, we only tackle the adaptation on corpus level. In future we are going to explore LM adaptation on document and sentence level. Besides, we also intend to apply adaptation to multiple LMs. Although our method works well on most dataset pairs, there still exist some pairs on which our method fails. Therefore, it will be interesting to further investigate the factors that determine the adaptation performance.

Acknowledgments

We thank Shujie Liu for his 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

Amittai Axelrod, Xiaodong He and Jianfeng Gao. 2011. Domain Adaptation via Pseudo In-Domain Data Selection. *In Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, 355-362, Edinburgh, July, 2011.

David Chiang. 2005. A Hierarchical Phrase-based Model for Statistical Machine Translation. *In Proceedings of the 43rd Annual Meeting of the ACL*, 263-270.

Almut Hildebrand, Matthias Eck, Stephan Vogel, and Alex Waibel. 2005. Adaptation of the Translation Model for Statistical Machine translation based on Information Retrieval. *In Proceedings of EAMT*, Budapest, Hungary.

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

George Foster, Cyril Goutte and Roland Kuhn. 2010. Discriminative Instance Weighting for Domain Adaptation in Statistical Machine Translation. *In Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, 451-459, MIT, Massachusetts, USA, October 2010.

Philipp Koehn and Josh Schroeder. 2007. Experiments in Domain Adaptation for Statistical Machine Translation. *In Proceedings of the Second Workshop on Statistical Machine Translation*, 224-227, Prague, June 2007.

Mu Li, Yinggong Zhao, Dongdong Zhang and Ming Zhou. 2010. Adaptive Development Data Selection for Log-linear Model in Statistical Machine Translation. *In Proceedings of the 23rd International Conference on Computational Linguistics (Coling 2010)*, 662-670.

Yajuan Lü, Jin Huang and Qun Liu. 2007. Improving Statistical Machine Translation Performance by Training Data Selection and Optimization. *In Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, 343-350.

Spyros Matsoukas, Antti-Veikko Rosti and Bing Zhang. 2009. Discriminative Corpus Weight Estimation for Machine Translation. *In Proc. of the Conference on Empirical Methods in Natural Language Processing*, 160-167

Behrang Mohit, Frank Liberato and Rebecca Hwa. 2009. Language Model Adaptation for Difficult To Translate Phrases. *In Proceedings of the 13th Annual Conference of the EAMT*, 160-167.

Behrang Mohit, Rebecca Hwa and Alon Lavie. 2010. Using Variable Decoding Weight for Language Model in Statistical Machine Translation *In The Proceedings of the 9th Conference of the Association for Machine Translation in the Americas*, Colorado

Robert C. Moore and William Lewis. 2010. Intelligent Selection of Language Model Training Data. *In Proceedings of the ACL 2010 Conference Short Papers*, 220-224, Uppsala, Sweden.

Franz Och. 2003. Minimum Error Rate Training in Statistical Machine Translation. *In Proceedings of the 41th Annual Meeting of the Association for Computational Linguistic (ACL)*, Sapporo, Japan.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a Method for Automatic Evaluation of Machine Translation. *In Proceedings of the 40th Annual Meeting of the Association for Computational Linguistic (ACL)*, 311-318.

Matthew Snover, Bonnie Dorr, Richard Schwartz, Linnea Micciulla and John Makhoul. 2006. A Study of Translation Edit Rate with Targeted Human Annotation. *In Proceedings of Association for Machine Translation in the Americas*

Yuta Tsuboi, Hisashi Kashima, Shohei Hido, Steffen Bickel and Masashi Sugiyama. 2008. Direct Density Ratio Estimation for Large-scale Covariate Shift Adaptation. *In Proceedings of the Eighth SIAM International Conference on Data Mining*, pp. 443–454, 2008.

Bing Zhao, Matthias Eck and Stephan Vogel. 2004. Language Model Adaptation for Statistical Machine Translation with Structured Query Models. *In Proceedings of International Conference on Computational Linguistics(COLING)*, Geneva, August.