Automatic Tune Set Generation for Machine Translation with Limited In-domain Data

Jinying Chen, Jacob Devlin, Huaigu Cao, Rohit Prasad, and Premkumar Natarajan
Raytheon BBN Technologies
10 Moulton Street, Cambridge 02138, USA
{jchen, jdevlin, hcao, rprasad, pnatarajan}@bbn.com

Abstract

Many effective adaptation techniques for statistical machine translation crucially rely on in-domain development sets to learn model parameters. In this paper we present a novel method that automatically generates the matching tune set for Arabic-to-English MT with limited in-domain data[1]. This technique improves our MT system over two baselines (tuned on data from the same domain but different genres) by 1.2 and 3.5 BLEU points using significantly less tuning data (1/6 and 1/2 of the baselines). Lexical and morphological features contribute to the success of our method in different ways. Generating tune sets using length distribution also improves the system significantly. Finally, our method obtains competitive results in experiments where in-genre tune sets are available.

1 Introduction

Adapting statistical machine translation (SMT) systems to different domains is a well-known and challenging problem. Many effective techniques developed for SMT adaptation crucially rely on in-domain development sets to learn model parameters or interpolation weights. For example, Koehn and Schroeder (2007); Ueffing *et al*., (2007); Matsoukas *et al*. (2009); Foster *et al*., (2010), to name a few. However, in some situations, in-domain data can be so limited and in a few cases, no matching tune sets are available. Our problem falls into this category.

Most existing work in domain adaptation for SMT focused on language models, translation models, lexicons and parallel training data (Koehn and Schroeder, 2007; Lü *et al.*, 2007; Wu *et al.*, 2008; Matsoukas *et al.*, 2009; Foster *et al.*, 2010). Tune set adaptation shares a belief with other adaptation techniques that using training data similar to the test set (in domain, topic, and style) plays a critical role in SMT performance. A unique feature of this problem, however, is the high demands of matching quality. Many parameters in the SMT system are estimated using the tune set, so negative effects caused by noise (e.g., mismatch in topic and translation style) can be propagated easily. In fact, a large number of SMT domain adaptation techniques also adopted a general framework that requires a tune set to learn model parameters (Ueffing *et al.*, 2007) or interpolation weights for data from different domains (Koehn and Schroeder, 2007; Matsoukas *et al.*, 2009; Foster *et al.*, 2010).

In this paper we present a highly effective method that automatically generates matching tune sets for an Arabic-to-English MT task with considerably limited in-domain data. Our method is based on the nearest neighbor approach and a novel *n*-gram based similarity metric. It generates the tune set by extracting the nearest neighbors from a data set of mixed, different genres for each test segment. This method can be applied to any new test set because it only uses the source side of the test segments to find neighbors. Word based and morphological tag based features were used to capture different similarity patterns between neighbors. Compared with two baseline systems, which were tuned on the full data set and one of its subsets, the MT system tuned on the automatically generated tune set increased the BLEU scores by 1.2 and 3.5 points (29.66 vs. 28.43 and 29.66 vs. 26.11), respective-

[1] This paper is based upon work supported by the DARPA MADCAT Program (Approved for Public Release, Distribution Unlimited). The views expressed are those of the authors and do not reflect the official policy or position of the Department of Defense or the U.S. Government.

ly. Furthermore, the tune sets generated by our method are much compact at only 1/6 and 1/2 the size of the baseline tune sets, respectively.

Further experiments suggested that both lexical and morphological features contributed to the effectiveness of this method. Length distribution is another important factor that affected performance. By using a length penalty score, our method naturally captured length distribution of the test set. Two comparative experiments with matching in-domain tune sets also obtained competitive results, which confirmed the robustness of our method. Another contribution of our work is to provide empirical evidence for various factors that impact tune set quality.

The rest of this paper is organized as follows. We reviewed related work in Section 2. In Section 3, we introduce our translation problem and the specific difficulty we faced. The similarity measure and features used by our tune set generation method are discussed in Section 4. In Section 5, we introduce the general techniques we used to adapt our MT system to the new domain. Experimental setup is described in Section 6 and experimental results and discussions are provided in Section 7. We conclude our paper in Section 8.

2 Related Work

Utiyama *et al.*, (2009) used a nearest neighbor-based approach to find optimal tune sets from a relatively large amount of in-domain parallel training corpora. Their method used the average of BLEU-1 to BLEU-4 scores to measure segment-level similarity. It outperformed a random sampling-based baseline by over 2 points in BLEU. Unlike their work, we developed a new similarity metric by observing the "*bias nature*" of BLEU in measuring segment-level similarity. Our experiments showed that our method was more effective in finding the matching tune set.

Hui *et al.* (2010) described the strategy for choosing the best tune set from a list of available in-domain tune sets based on their similarities to the test set (measured by a modified BLEU score). Unlike their work, we constructed the tune set from scratch by using a segment-level similarity measure.

Apart from tune set sampling and selection techniques mentioned above, some attempts have been made in sampling parallel training data. Lü *et al.*, (2007) used the nearest neighbor-based method to generate a compact parallel training corpus that matched the test and tune sets. They used the standard TF-IDF weighting scheme to measure segment-level similarity. They observed that, over a threshold (1000 in their case) of the number of neighbors used, the MT performance would drop due to noisy data included. We observed similar phenomena in our experiments (as discussed in Section 7.3) but the threshold was much lower (=2 in our case). This suggests that accurate matching is more demanding in tune set generation than in training set generation

3 Problem Setting

Our task is Arabic-to-English translation on image text from the field (legal filings, *etc.*), which we will refer to as the *Field Document* domain. This task has limited in-domain data, with 0.4M translated words in total. We have a state-of-the-art MT system trained on a large amount of out-of-domain data, including 50M words of newswire and web bilingual data and 9 billion words of English text (to train the language model). This is a typical domain adaptation problem.

A specific difficulty we faced in this task, however, is that the small size in-domain data was further divided into three genres: handwritten (HW), machine print (MP) and mixed-form (MX). The three genres have overlap in topics but are quite different in style. HW data are mainly fluent text and long sentences; MP data were extracted from printed forms and are mainly short phrases or segmented (diffluent) text; MX data were extracted from different forms with both printed and handwritten text, and are a more balanced mixture of fluent and diffluent text. Genre information was given at both document-level and segment level. On average, an HW document has over 95% HW segments, an MP document has over 85% MP segments, and an MX document is more balanced, but still has over 65% MX segments[2]. It was required to report translation scores on each genre at document-level separately. Furthermore, the document distribution for these three genres is extremely unbalanced: 1929 HW documents, 590 MX documents and only 68 MP documents. Since MP data was so limited, we reserved all of them as the MP test set to ensure the reliability of the testing results.

In sum, our task is to build MT systems for three genres with limited in-domain data, one of which is completely missing its genre-matched training data.

[2] The MX segments cannot be automatically divided into handwritten and printed parts for translation purpose.

4 Nearest Neighbor Based Automatic Tune Set Generation

To automatically generate the matching tune set for the MP data, we used a nearest neighbor approach which was inspired by Utiyama *et al.* (2009). However, we developed a novel similarity metric and exploited different *n*-gram features, which we believe better fit our problem. This was confirmed by our experimental results.

4.1 Similarity Metric

We defined a similarity metric that looks like BLEU (Papineni *et al.*, 2002) but is significantly different in nature.

$$match_i(c,t) = \frac{\sum_{\{n_i \in c \cap t\}} \min(count_c(n_i), count_t(n_i))}{\sum_{\{n_i \in t\}} count_t(n_i)} \quad (1)$$

$$sim(c,t) = -\frac{|len(c) - len(t)|}{len(t)} + \frac{1}{N}\sum_{i=1}^{N} \log(match_i(c,t)) \quad (2)$$

where t is a given test segment and c is the candidate segment. n_i is any i order n-gram. $count_x(n_i)$ is the number of occurrences of n_i in segment x. $match_i(c,t)$ looks like the precision score of BLEU but we treat c as the "reference" and t as the "hypothesis"[3]. So unlike BLEU precision, this score is not affected by the length of a candidate segment.

len(x) is the number of occurrences of *1*-gram's in segment x. $-\frac{|len(c) - len(t)|}{len(t)}$ is the length penalty score, which penalizes the longer and shorter candidates equally. $sim(c,t)$ is the similarity measure which combines the length penalty score and the n-gram matching scores in a way similar to BLEU. N is the highest order of n-grams used (N=4 in our case).

The major difference between our measure and BLEU is that it uses only a symmetric length penalty score to enforce length matching, while BLEU relies on its precision score to penalize longer hypotheses and a non-symmetric length penalty score to penalize shorter ones. Simple mathematical calculation shows that BLEU, by its nature, favors longer hypotheses (i.e., candidate segments) than shorter ones when they have equal numbers of overlapping n-grams with the reference (the given test segment) and their length distances from the test segment are equal. This bias is not a big issue when measuring similarity among blocks of text but can be a problem when measuring segment-level similarity. This is why we designed a new similarity metric that handles length penalty in a different way.

Since it is likely to get zero-valued $match_i$'s at segment level, which will make their log values negative infinite, we uses a non-parametric approach to smooth our n-gram matching measure by adding 1 to the numerator and denominator of equation (1), as in equation (1)'.

$$match_i(c,t) = \frac{1 + \sum_{\{n_i \in c \cap t\}} \min(count_c(n_i), count_t(n_i))}{1 + \sum_{\{n_i \in t\}} count(n_i)} \quad (1')$$

We compared the length distribution of the tune sets generated by our method and by a BLEU-based similarity measure (Utiyama *et al.*, 2009). As shown in Fig. 1, the length distribution curve generated by our method (**MP-AG**, grey solid line) had less fluctuations than that generated by **BLEU** (black dotted line), compared with the curve of the MP test set (**MP-test**). Though length distribution is only one factor that impacts MT performance (to be discussed in Section 7.2), it gives us a clue that our measure is likely to achieve better MT performance (which was confirmed by our MT experiments).

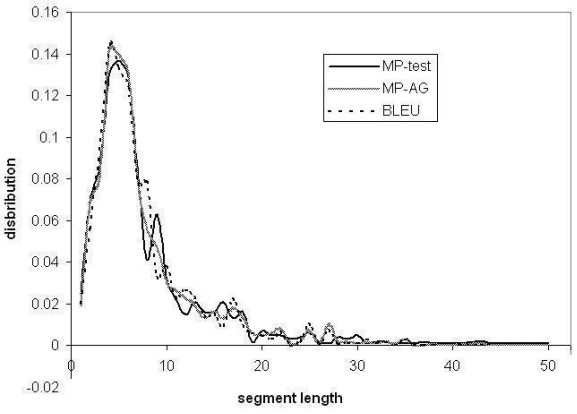

Figure 1. Length Distributions of the MP test set (MP-test), the tune sets generated by our method (MP-AG) and generated by BLEU

4.2 *N*-gram Features

We used two types of *n*-gram features: lexical based and morphological based.

Lexical *n*-grams are strong indicators for text similarity, which were used by many previous

[3] In practice, we omit the denominator of this item when ranking neighbors for a given test segment.

works to measure segment-level or data set level similarity (Lü *et al.*, 2007; Utiyama *et al.*, 2009; Hui *et al.*, 2010). Intuitively, lexical *1*-gram's and *2*-gram's are good indicators of topical similarity and higher order *n*-grams (*n>2*) are more responsible to capture similarity in styles. We extracted lexically-based *n*-grams from the source side (Arabic text without tokenization) of each bilingual sentence-pair.

One issue with using lexical *n*-grams is that only exact matches are counted. In order to have a more generalized model, we also compute the similarity score using morphological tags. Arabic is a morphological rich language, so its morphological tags hopefully can provide us a good balance between accuracy and generalization.

We used Sakhr Morphological Analyzer, a proprietary rule-based software, to generate the morphological tags for Arabic. The Sakhr tags are similar to English part-of-speech tags but have richer information about a word. For example, a tag for an Arabic verb may indicate tense, number, gender and voice. We kept all this information in a tag (i.e., did not generalize further) when matching morphological *n*-gram's.

Though less accurate, the generalization helps to capture more aspects in style similarity. For example, many MP segments contain names, dates and numbers. By using morphological features, our method can discover segments that share the same sentence structures with an MP segment but do not necessarily contain the same names or numbers.

We used these two types of features independently. That is, we always find 2x*n* (*n*=1 in our experiments) nearest neighbors, *n* by lexical features and *n* by morphological features.

4.3 Treatment of Duplicate Neighbors

Since the nearest neighbors were extracted for each test segment, the resulting tune set had duplicate segments. We kept duplicate instances because the number of duplicates naturally reflected to which degree a selected segment fit the whole test set. In the real implementation, we refined our MT system to support segment level weighting for tune sets. That is, we used the non-duplicate tune set with its segments weighted by the number of their duplicates. This sped up the training procedure, especially when there were many duplicates in the tune set or the system need to be tuned for much iteration. In Section 7, we only report the size of non-duplicate tune sets for all the experiments.

5 MT System Description

5.1 Baseline MT System

We used a state-of-the-art hierarchical decoder in our experiments (Shen *et al.*, 2008). The features it uses in decoding and n-best rescoring includes a small set of linguistic and contextual features, such as word translation probabilities, rule translation probabilities, language model scores, and target side dependency scores. In addition, it uses a large number of discriminatively tuned features, similar to those described in (Chiang *et al.* 2009). The system used a 3-gram language model (LM) for decoding and a 5-gram LM for rescoring. Both LMs were trained on billions of words of English text in news and web blogs. Feature scores are combined with a log-linear model. The feature weights were set by optimizing the BLEU score on the tune set.

5.2 Domain Adaptation

The general framework we used to adapt our baseline MT system to the new domain follows the line of Koehn and Schroeder (2007). We trained a separate language model using the target side of our in-domain parallel training data, and discriminatively estimated the interpolation weight with the standard language model. To adapt the translation model, we discriminatively estimate separate feature weights and penalties for rules extracted from the in-domain and out-of-domain parallel training.

This adaptation procedure improved the results on the HW test set by 8 points of BLEU and TER (see Table 1). We used this system in all the experiments on tune set generation.

Condition	BLEU	TER
Train: News Tune: News	19.99	61.58
Train: News+Field Tune: Field	28.23	53.33

Table 1. Baseline scores before/after adaptation

It is worth noting that the MT systems we developed will be applied on the output from a state-of-the-art optical character recognition (OCR) system. Because the OCR errors usually reduce MT performance significantly, we only used the transcribed text to develop our MT system and applied the final system on the OCR output with all the system parameters fixed. Therefore, we reported our experimental results mainly on the transcribed text, except that we

provided the MT performance scores on the OCR input of the MP genre in order to show the gain from using our method was applicable to the noisy input from OCR.

6 Experimental Setting

6.1 Data Sets

As introduced in Section 3, in our problem, there was limited in-domain data in the *Field Document* domain and the document distribution for the three genres was unbalanced.

We reserved all 68 MP documents for the MP test set. To create a tune set for this genre, we randomly picked MP-labeled segments from 153 HW documents and 229 MX documents. This formed the first baseline in our automatic Tune set generation experiments. The second baseline was the single big tune set by merging the MP, HW and MX tune sets (called ALL-tune).

The test and tune sets for the MX and HW genres were randomly picked documents with the same genre labels. The remaining documents were used as the parallel training data for extracting in-domain translation rules and training the in-domain LMs. Table 2 summarizes our data set division.

Data Set	Num of segments	Source
MP-test	1,093	MP
HW-test	3,150	HW
MX-test	2,400	MX
MP-tune	1,876	MX,HW
HW-tune	2,730	HW
MX-tune	2,522	MX
All-tune	7,091[4]	HW, MX
Parallel-training	25,864	MX,HW

Table 2. In-domain Data Division

We used the same parallel training data in all the experiments described in this paper to compare the pure effects from different tune sets. In practice, after we determine the specific tune set for each genre, we can add all the unselected data to parallel training to maximize the gains.

6.2 Experimental Conditions

In the MP experiments, we compared MT performance using our method (**Auto-Gen**) with the following baseline conditions:

- **MP**: MP-tune
- **ALL**: All-tune
- **BLEU-1**: tune set extracted from All-tune by duplicating the method described in (Utiyama *et al.*, 2009); use lexical features only
- **BLEU-2**: same as BLEU-1; use both lexical and morphological features

To separate various factors that impact the effectiveness of our method, we compared four **Auto-Gen** conditions where the segment-level similarity was measured in different ways:

- **Len**: only use the length penalty measure in Eq. 2 to measure the segment similarity
- **Len+Lex**: use the full Eq. 2 but only use lexical based *n*-grams
- **Len+Mrf**: use the full Eq. 2 but only use morphological based *n*-grams
- **Len+Lex+Mrf**: our complete method

We also tested our method on the other two genres in the three conditions similar to the MP experiments: **Auto-Gen**, **HW/MX**, **ALL.** However, because HW-tune and MX-tune are in the same genre as their test sets, they are actually *upper-bound* in some sense rather than baselines. **ALL** is also harder to beat because 1/3~2/5 tune-ALL segments are from the same genre as the HW (or MX) test set. Nevertheless, the results on these two genres can add evidence on how well our method works.

7 Results

Table 3 showed the results on the MP test set using automatic tune set generation. The system using our complete method (**Len+Lex+Mrf**) outperformed the system tuned on the MP tune set (**MP**) by 3.5 points in BLEU and 3 points in TER. Furthermore, the automatically generated tune set was more compact, with its size only about half of the MP tune set. Compared with using all the tuning data (**ALL**), our method achieved 1.2 points gain in the BLEU score and 0.9 point gain in TER. This gain was also significant, especially when considering that it only used about 1/6 of all the tuning data.

Surprisingly, MT performance using the MP tune set (**MP**), which was composed of MP segments from the HW and MX genres, was significantly lower than using all the tuning data (**ALL**). Further data analysis suggested that the unmatched length distribution between the MP tune set and the MP test set and the low vocabulary

[4] The MP-tune and HW-tune sets have a small portion of overlap, so the number of segments in All-tune is slightly different from the sum of MP-tune, HW-tune and MX-tune.

coverage were the two culprits for the performance drop. The lesson learned here is we should not fully trust segment-level genre labels to find a matching tune set. We will discuss this in greater details in Sections 7.1 and 7.2.

We further compared our method with the method (**BLEU-1)** as described by Utiyama *et al*. (2009). They used the averaged BLEU-*i* scores (*i* =1, 2, 3, 4) to measure segment-level similarity and extracted 2 nearest neighbors for each test segment. The results showed that our method performed better, with 1.5 point gain in BLEU and 1.2 point gain in TER.

To verify the appropriateness of the various considerations we had in designing our similarity measure, we compared our method with another method (**BLEU-2**) that used the averaged BLEU-*i* scores (*i* =1,2,3,4) as the similarity measure and used the same lexical and morphological *n*-gram features as ours. Compared with **BLEU-2**, our method had 1.1 point gain in BLEU and the same TER value. **BLEU-2** is better than **BLEU-1** (0.54 point gain in BLEU and 1.2 point gain in TER), suggesting that using morphological features is helpful. We will have more discussions in this aspect in Section 7.2.

Tune Set	Num Segs	BLEU	TER
MP	1,876	26.11	54.81
ALL	7,091	28.43	52.76
Len+Lex+Mrf	1,081	29.66	51.82
BLEU-1	1,168	28.03	53.04
BLEU-2	1,084	28.57	51.89

Table 3. MT Performance on Transcribed MP Test Set Using Different Tune Sets

Further experiments confirmed that the MT system developed on the automatically generated MP tune set achieved consistent gains on the input with OCR errors (word error rate=9.4%), as shown in Table 4.

Tune Set	BLEU	TER
MP	24.26	57.60
ALL	26.24	56.12
Len+Lex+Mrf	27.11	55.23

Table 4. MT Performance on OCR output of MP Test Set by Using Different Tune Sets

7.1 Effect of Length Distribution

To investigate the significant performance drop by using the MP tune set, we compared the segment-level length distribution of this set and the MP test set. The difference was obvious (see Fig. 2, dotted line vs. black solid line). In contrast, the length distribution of the in-genre tune sets for the HW and MX data matched their test sets well (we omit the figures here due to space limits). This suggests that the MP segments from the HW and MX documents are significantly different from the MP test data.

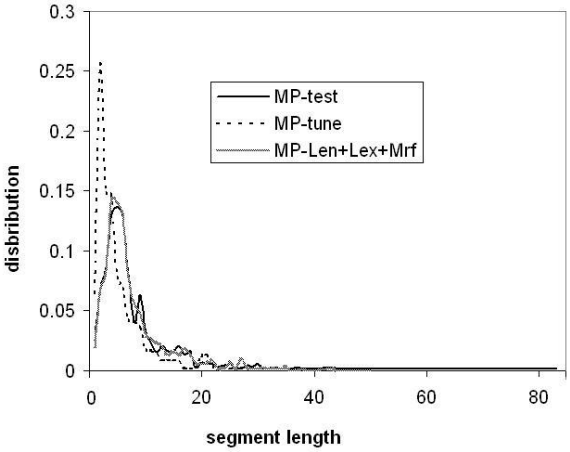

Figure 2. Length Distributions of MP-test, MP-tune and the tune set generated by Auto-Gen (Len+Lex+Mrf)

Intuitively, length distribution is a good indicator for the style of text from different sources. Therefore keeping similar length distribution is essential to getting a good matching tune set. Our similarity measure (as defined in Eq. 2) used a length penalty score to enforce length similarity among neighbors. The length distribution of the tune set generated by our method fit that of the MP test set very well (Fig. 2, grey line vs. black solid line). To separate the effect of this factor from other factors like *n*-gram based matching, we compared our method with a method that used only the length similarity (or penalty) scores to rank the neighbors of a test segment. For a fair comparison, we extracted two nearest neighbors for a test segment in both methods. If a test segment has more than two equally-nearest neighbors measured by length, we randomly picked two segments from them.

As expected, the length distribution of the tune sets generated by length-based sampling fit the MP test set well. The MT experiments (Table 5) showed that using the length similarity itself (**Len**) improved system performance by 1.4 points of BLEU and 1.1 points of TER scores over the **MP** baseline, but still significantly worse than using our complete method (**Len+Lex+Mrf**). These results suggest that

modeling length distribution is useful but by itself won't guarantee to find the best tune set.

Tune Set	Num Segs	BLEU	TER
MP	1,876	26.11	54.81
Len	1,338	27.58	53.79
Len+Lex+Mrf	1,081	29.66	51.82

Table 5. Effect of Length Distribution on Finding Matching Tune Sets

7.2 Lexical vs. Morphological *N*-grams

To separate the contributions from the lexical and morphological *n*-gram features, we compared our method with two other methods that used the same similarity measure but used only the lexical or the morphological features. The results (Table 6) showed that neither type of features (**Len+Lex** or **Len+Mrf**) was as effective as their combination (**Len+Lex +Mrf**) in improving the MT scores, though they all outperformed the **MP** baseline.

As discussed in Section 4.2, the lexical *n*-grams are expected to characterize topical similarity to a greater degree than the morphological features. To estimate the topical similarity among different data sets, we compared the out-of-vocabulary (OOV) rates[5] (against the MP test set) of the tune sets generated by the above three methods. As shown in Table 6, the tune set generated by lexical *n*-gram matching had smaller OOV rate than morphological *n*-gram matching (36.34 vs. 36.84). Combining them reduced the OOV rate by over 4 percent to 32.18. The higher OOV rate of the MP tune set (45.51) further suggests that this set is less similar to the MP test set.

7.3 Effect of Increasing Neighbors

Given that **Len+Lex** was better than **Len+Mrf** in both the OOV rate and the MT performance, one may question if using only lexical features and 2 nearest neighbors will be better. In fact, this method (**Len+2Lex** in Table 6**)** was worse than **Len+Lex**, though it had a lower OOV rate. One possible reason is the noise introduced by using more, but less similar, neighbors. Further experiments (comparing 2x*n*, *n*=1, 2, 3, 5, 10, nearest neighbors) showed that using more neighbors decreased the MT performance (Table 7). This result suggests a trade-off between precision (accurate matching) and recall (enlarging the vocabulary). Unlike training corpora creation where increasing vocabulary coverage had a privileged priority (Biçici and Yuret, 2011), accurate matching (similarity) is more important for tune set generation.

Tune Set	OOV (%)	Num Segs	BLEU	TER
MP	45.51	1,876	26.11	54.81
Len+Lex	36.34	682	28.16	52.99
Len+Mrf	36.84	616	27.18	52.70
Len+Lex +Mrf	32.18	1,081	29.66	51.82
Len+2Lex	32.98	1,176	27.32	52.62

Table 6. Effect of Lexical vs. Morphological Features on Finding the Matching Tune Sets

n	1	2	3	5	10
OOV	32.18	29.44	28.09	25.94	24.80
BLEU	29.66	28.78	28.31	27.99	27.75

Table 7. Effect of Increasing Neighbors (each experiment used 2x*n* nearest neighbors)

Tune Set	Num Segs	BLEU	TER
HW	2,730	28.23	53.33
ALL	7,091	28.09	52.85
Random	2,369	27.09	53.11
Len+Lex+Mrf	2,350	27.65	52.98

Table 8. MT Performance on Transcribed Handwritten Test Set Using Different Tune Sets

7.4 Experiments on HW and MX Test Sets

We also applied our tune set generation method on the HW and MX data. The results showed that the HW tune set (**HW**) outperformed our method (**Len+Lex+Mrf**) by 0.6 point BLEU and 0.4 point TER (Table 8) and the MX tune set (**MX**) outperformed by 0.9 point BLEU and 0.6 point TER (Table 9). The within 1 point performance drop was acceptable since the HW and MX tune sets, which were randomly picked from the same genres as their test sets, were similar to their test sets already (measured by the length distribution and the OOV rates). Comparing with the randomly generated tune sets (**Random**) in the same size, our method improved the MT performance by 0.6 BLEU points on the HW test set and 0.7 BLEU points on the MX test set.

Comparing with using all the tuning data (**ALL**), our method achieved close performance (within 0.1~0.4 points in BLEU and TER) while using much less data (1/4~1/3). The total amount

[5] The OOV rate numbers are high because the lexicons generated from the tune sets (several thousand segments) are small.

of CPU time required to run tuning is thus reduced to 1/4~1/3 of the original cost, since this time is directly proportional to the size of the tune set. The time required to run our tune set selection procedure is well over 100x faster than the tuning itself, so it is not a significant factor in the total run time. This added further evidence to the robustness and effectiveness of our method.

Tune Set	Num Segs	BLEU	TER
MX	2,522	37.05	42.95
ALL	7,091	36.48	43.71
Random	1,808	35.44	44.41
Len+Lex+Mrf	1,790	36.12	43.52

Table 9. MT Performance on Transcribed Mixed Test Set Using Different Tune Sets

8 Conclusions

This paper presents a novel method to automatically generate matching tune sets for MT tasks with limited in-domain data. With this method our MT system achieved significantly better performance (measured by BLEU and TER scores) than two baseline systems using significantly less tuning data. The performance gains were consistent on input text with OCR errors. This method also achieved competitive results on two other MT tasks with in-genre tune sets. In addition, we provide empirical evidence that length distribution modeling, lexical and morphological *n*-gram matching are all important factors contributing to the success of our method. They were able to capture topical and style similarities in different ways. We also showed that, compared with parallel training data extraction and generation, precision (accurate matching) was more important than recall (increasing vocabulary coverage). In the future, we hope to extend this method to training data creation for MT with limited in-domain data in an active learning framework.

References

Ergun Bicici and Deniz Yuret. 2011. Instance selection for machine translation using feature decay algorithms. In *Proceedings of the EMNLP 2011 Sixth Workshop on Statistical Machine Translation*, Edinburgh, England, July.

David Chiang, Kevin Knight, and Wei Wang. 2009. 11,001 new features for statistical machine translation. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the ACL*, pages 218–226.

Hui Cong, Zhao Hai, Lu Bao-Liang, and Song Yan. 2010. An empirical study on development set selection strategy for machine translation learning. In *Proceedings of the Joint Fifth Workshop on Statistical Machine Translation and MetricsMATR*, pages 42–46, Uppsala, Sweden, July. Association for Computational Linguistics.

George Foster and Roland Kuhn. 2009. Stabilizing minimum error rate training. In *Proceedings of the 4th Workshop on Statistical Machine Translation(WMT)*, Boulder, Colorado, USA.

Philipp Koehn and Josh Schroeder. 2007. Experiments in domain adaptation for statistical machine translation. In *Proceedings of the Second Workshop on Statistical Machine Translation*, pages 224–227.

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*, pages 343–350.

Spyros Matsoukas, Antti-Veikko I. Rosti, and Bing Zhang. 2009. Discriminative corpus weight estimation for machine translation. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Singapore.

Kishore Papineni, Salim Roukos, ToddWard, andWei-Jing Zhu. 2002. BLEU: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318.

Libin Shen, Jinxi Xu, and Ralph Weischedel. 2008. A new string-to-dependency machine translation algorithm with a target dependency language model. In *Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 577–585.

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

Masao Utiyama, Hirofumi Yamamoto, and Eiichiro Sumita. 2009. Two methods for stabilizing MERT: NICT at IWSLT 2009. In *Proceedings of International Workshop on Spoken Language Translation(IWSLT)*, Tokyo, Japan.

Hua Wu, Haifeng Wang, and Chengqing Zong. 2008. Domain adaptation for statistical machine translation with domain dictionary and monolingual corpora. In *Proceedings of the 22nd International Conference on Computational Linguistics (Coling 2008)*, pages 993–1000. Manchester, USA.