Translation Model Generalization using Probability Averaging for Machine Translation

Nan Duan[1], **Hong Sun**
School of Computer Science and Technology
Tianjin University
v-naduan@microsoft.com
v-hongsun@microsoft.com

Ming Zhou
Microsoft Research Asia
mingzhou@microsoft.com

Abstract

Previous methods on improving translation quality by employing multiple SMT models usually carry out as a second-pass decision procedure on hypotheses from multiple systems using extra features instead of using features in existing models in more depth. In this paper, we propose *translation model generalization* (TMG), an approach that updates probability feature values for the translation model being used based on the model itself and a set of auxiliary models, aiming to enhance translation quality in the first-pass decoding. We validate our approach on translation models based on auxiliary models built by two different ways. We also introduce novel probability variance features into the log-linear models for further improvements. We conclude that our approach can be developed independently and integrated into current SMT pipeline directly. We demonstrate BLEU improvements on the NIST Chinese-to-English MT tasks for single-system decodings, a system combination approach and a model combination approach.

1 Introduction

Current research on Statistical Machine Translation (SMT) has made rapid progress in recent decades. Although differed on paradigms, such as phrase-based (Koehn, 2004; Och and Ney, 2004), hierarchical phrase-based (Chiang, 2007) and syntax-based (Galley *et al.*, 2006; Shen *et al.*, 2008; Huang, 2008), most SMT systems follow the similar pipeline and share common translation probability features which constitute the principal components of translation models. However, due to different model structures or data distributions, these features are usually assigned with different values in different translation models and result in translation outputs with individual advantages and shortcomings.

In order to obtain further improvements, many approaches have been explored over multiple systems: system combination based on confusion network (Matusov *et al.*, 2006; Rosti *et al.*, 2007; Li *et al.*, 2009a) develop on multiple *N*-best outputs and outperform primary SMT systems; consensus-based methods (Li *et al.*, 2009b; DeNero *et al.*, 2010), on the other hand, avoid the alignment problem between translations candidates and utilize *n*-gram consensus, aiming to optimize special decoding objectives for hypothesis selection. All these approaches act as the second-pass decision procedure on hypotheses from multiple systems by using extra features. They begin to work only after the generation of translation hypotheses has been finished.

In this paper, we propose *translation model generalization* (TMG), an approach that takes effect during the first-pass decoding procedure by updating translation probability features for the translation model being used based on the model itself and a set of auxiliary models. Bayesian Model Averaging is used to integrate values of identical features between models. Our contributions mainly include the following 3 aspects:

- *Alleviate the model bias problem based on translation models with different paradigms.* Because of various model constraints, translation models based on different paradigms could have individual biases. For instance, phrase-based models prefer translation pairs with high frequencies and assign them high

[1] This work has been done while the author was visiting Microsoft Research Asia.

Proceedings of the 23rd International Conference on Computational Linguistics (Coling 2010), pages 304–312,
Beijing, August 2010

probability values; yet such pairs could be disliked or even be absent in syntax-based models because of their violation on syntactic restrictions. We alleviate such model bias problem by using the generalized probability features in first-pass decoding, which computed based on feature values from all translation models instead of any single one.

- *Alleviate the over-estimation problem based on translation models with an identical paradigm but different training corpora.*
  In order to obtain further improvements by using an existing training module built for a specified model paradigm, we present a random data sampling method inspired by bagging (Breiman, 1996) to construct translation model ensembles from a unique data set for usage in TMG. Compared to results of TMG based on models with different paradigms, TMG based on models built in such a way can achieve larger improvements.

- *Novel translation probability variance features introduced.*
  We present how to compute the variance for each probability feature based on its values in different involved translation models with prior model probabilities. We add them into the log-linear model as new features to make current SMT models to be more flexible.

The remainder of this paper is organized as follows: we review various translation models in Section 2. In Section 3, we first introduce Bayesian Model Averaging method for SMT tasks and present a generic TMG algorithm based on it. We then discuss two solutions for constructing TM ensembles for usage in TMG. We next introduce probability variance features into current SMT models as new features. We evaluate our method on four state-of-the-art SMT systems, a system combination approach and a model combination approach. Evaluation results are shown in Section 4. In Section 5, we discuss some related work. We conclude the paper in Section 6.

2 Summary of Translation Models

Translation Model (TM) is the most important component in current SMT framework. It provides basic translation units for decoders with a series of probability features for model scoring. Many literatures have paid attentions to TMs from different aspects: DeNeefe *et al.* (2007) compared strengths and weaknesses of a phrase-based TM and a syntax-based TM from the *statistic* aspect; Zollmann *et al.* (2008) made a systematic comparison of three TMs, including phrasal, hierarchical and syntax-based, from the *performance* aspect; and Auli *et al.* (2009) made a systematic analysis of a phrase-based TM and a hierarchical TM from the *search space* aspect.

Given a word-aligned training corpus, we separate a TM training procedure into two phases: *extraction phase* and *parameterization phase*.

Extraction phase aims to pick out all valid translation pairs that are consistent with predefined model constraints. We summarize current TMs based on their corresponding model constraints into two categories below:

- *String-based* TM (string-to-string): reserves all translation pairs that are consistent with word alignment and satisfy length limitation. SMT systems using such TMs can benefit from a large convergence of translation pairs.

- *Tree-based* TM (string-to-tree, tree-to-string or tree-to-tree): needs to obey syntactic restrictions in one side or even both sides of translation candidates. The advantage of using such TMs is that translation outputs trend to be more syntactically well-formed.

Parameterization phase aims to assign a series of probability features to each translation pair. These features play the most important roles in the decision process and are shared by most current SMT decoders. In this paper, we mainly focus on the following four commonly used dominant probability features including:

- translation probability features in two directions: $p(\bar{e}|\bar{f})$ and $p(\bar{f}|\bar{e})$
- lexical weight features in two directions: $p_{lex}(\bar{e}|\bar{f})$ and $p_{lex}(\bar{f}|\bar{e})$

Both string-based and tree-based TMs are state-of-the-art models, and each extraction approach has its own strengths and weaknesses comparing to others. Due to different predefined model constraints, translation pairs extracted by different models usually have different distributions, which could directly affect the resulting probability feature values computed in parame-

terization phase. In order to utilize translation pairs more fairly in decoding, it is desirable to use more information to measure the quality of translation pairs based on different TMs rather than totally believing any single one.

3 Translation Model Generalization

We first introduce Bayesian Model Averaging method for SMT task. Based on it, we then formally present the generic TMG algorithm. We also provide two solutions for constructing TM ensembles as auxiliary models. We last introduce probability variance features based on multiple TMs for further improvements.

3.1 Bayesian Model Averaging for SMT

Bayesian Model Averaging (BMA) (Hoeting *et al.*, 1999) is a technique designed to solve uncertainty inherent in model selection.

Specifically, for SMT tasks, f is a source sentence, $\mathcal{D}$ is the training data, $\mathcal{M}_k$ is the k^{th} SMT model trained on $\mathcal{D}_k \subset \mathcal{D}$, $p_k(\cdot|f,e)$ represents the probability score predicted by $\mathcal{M}_k$ that f can be translated into a target sentence e. BMA provides a way to combine decisions of all $K+1$ SMT models by computing the final translation probability score $\bar{p}_E(\cdot|f,e,\mathcal{D})$ as follows:

$$\bar{p}_E(\cdot|f,e,\mathcal{D}) = \sum_{k=0}^{K} p(\mathcal{M}_k|\mathcal{D}_k)\, p_k(\cdot|f,e), \quad (1)$$

where $p(\mathcal{M}_k|\mathcal{D}_k)$ is the prior probability that $\mathcal{M}_k$ is a true model. For convenience, we will omit all symbols $\mathcal{D}_k$ in following descriptions.

Ideally, if all involved models $\{\mathcal{M}_0, \dots, \mathcal{M}_K\}$ share the same search space, then translation hypotheses could only be differentiated in probability scores assigned by different SMT models. In such case, BMA can be straightly developed on the whole SMT models in either span level or sentence level to re-compute translation scores of hypotheses for better rankings. However, because of various reasons, e.g. different pruning methods, different training data used, different generative capabilities of SMT models, search spaces between different models are always not identical. Thus, it is intractable to develop BMA on the whole SMT model level directly.

As a tradeoff, we notice that translation pairs between different TMs share a relatively large convergence because of word length limitation. So we instead utilize BMA method to multiple TMs by re-computing values of probability features between them, and we name this process as translation model generalization.

3.2 A Generic BMA-based TMG Algorithm

For a translation model $\mathcal{M}_0$, TMG aims to re-compute its values of probability features based on itself and K collaborative TMs $\{\mathcal{M}_1, \dots, \mathcal{M}_K\}$. We describe the re-computation process for an arbitrary feature $p(\cdot|\bar{f},\bar{e}) \in \mathcal{M}_0$ as follows:

$$\bar{p}_E(\cdot|\bar{f},\bar{e}) = \sum_{k=0}^{K} p(\mathcal{M}_k)\, p_k(\cdot|\bar{f},\bar{e}), \quad (2)$$

where $p_k(\cdot|\bar{f},\bar{e})$ is the feature value assigned by $\mathcal{M}_k$. We denote $\mathcal{M}_0$ as the ***main model***, and other collaborative TMs as ***auxiliary models***. Figure 1 describes an example of TMG on two TMs, where the main model is a phrasal TM.

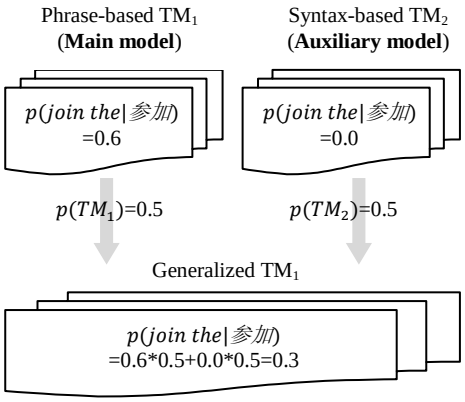

Figure 1. TMG applied to a phrasal TM (main model) and a syntax-based TM (auxiliary model). The value of a translation probability feature $p(join\ the|参加)$ in TM_1 is de-valued (from 0.6 to 0.3), in which 'join the' is absent in TM_2 because of its bad syntactic structure.

Equation 2 is a general framework that can be applied to all TMs. The only limitation is that the segmentation (or tokenization) standards for source (or target) training sentences should be identical for all models. We describe the generic TMG procedure in Algorithm 1[2].

[2] In this paper, since all data sets used have relative large sizes and all SMT models have similar performances, we heuristically set all $p(\mathcal{M}_k)$ equally to $1/(K+1)$.

Algorithm 1: TMG for a main model $\mathcal{M}_0$

for the k^{th} auxiliary TM **do**
  run training procedure on $\mathcal{D}_k$ with specified model constraints and generate $\mathcal{M}_k$
end for
for each translation pair $< \bar{f}, \bar{e} >$ in $\mathcal{M}_0$ **do**
  for each probability feature $p(\cdot\,|\bar{f}, \bar{e})$ **do**
    for each translation model $\mathcal{M}_k$ **do**
      if $< \bar{f}, \bar{e} >$ is contained in $\mathcal{M}_k$ **then**
        $\bar{p}_E(\cdot\,|\bar{f}, \bar{e}) \mathrel{+}= p(\mathcal{M}_k) p_k(\cdot\,|\bar{f}, \bar{e})$
      end if
    end for
  end for
end for
return the generalized $\mathcal{M}_0$ for SMT decoding

3.3 Auxiliary Model Construction

In order to utilize TMG, more than one TM as auxiliary models is needed. Building TMs with different paradigms is one solution. For example, we can build a syntax-based TM as an auxiliary model for a phrase-based TM. However, it has to re-implement more complicated TM training modules besides the existing one.

In this sub-section, we present an alternative solution to construct auxiliary model ensembles by using the existing training module with different training data extracted from a unique data set. We describe the general procedure for constructing K auxiliary models as follows:

1) Given a unique training corpus $\mathcal{D}$, we randomly sample $N\%$ bilingual sentence pairs without replacement and denote them as $\mathcal{D}_i$. N is a number determined empirically;
2) Based on $\mathcal{D}_i$, we *re-do* word alignment and train an auxiliary model $\mathcal{M}_i$ using the existing training module;
3) We execute Step 1 and Step 2 iteratively for K times, and finally obtain K auxiliary models. The optimal setting of K for TMG is also determined empirically.

With all above steps finished, we can perform TMG as we described in Algorithm 1 based on the K auxiliary models generated already.

The random data sampling process described above is very similar to bagging except for it not allowing replacement during sampling. By making use of this process, translation pairs with low frequencies have relatively high probabilities to be totally discarded, and in resulting TMs, their probabilities could be zero; meanwhile, translation pairs with high frequencies still have high probabilities to be reserved, and hold similar probability feature values in resulting TMs comparing to the main model. Thus, after TMG procedure, feature values could be smoothed for translation pairs with low frequencies, and be stable for translation pairs with high frequencies. From this point of view, TMG can also be seen as a TM smoothing technique based on multiple TMs instead of single one such as Foster *et al.* (2006). We will see in Section 4 that TMG based on TMs generated by both of these two solutions can improve translation quality for all baseline decoders on a series of evaluation sets.

3.4 Probability Variance Feature

The re-computed values of probability features in Equation 2 are actually the feature expectations based on their values from all involved TMs. In order to give more statistical meanings to translation pairs, we also compute their corresponding feature variances based on feature expectations and TM-specified feature values with prior probabilities. We introduce such variances as new features into the log-linear model for further improvements. Our motivation is to quantify the differences of model preferences between TMs for arbitrary probability features.

The variance for an arbitrary probability feature $p(\cdot) \in \mathcal{M}_0$ can be computed as follows:

$$p_V(\cdot) = \sum_{k=0}^{K} \{p_k(\cdot) - \bar{p}_E(\cdot)\}^2 p_k(\mathcal{M}_k), \qquad (3)$$

where $\bar{p}_E(\cdot)$ is the feature expectation computed by Equation 2, $p_k(\cdot)$ is the feature value predicted by $\mathcal{M}_k$, and $p_k(\mathcal{M}_k)$ is the prior probability for $\mathcal{M}_k$. Each probability feature now corresponds to a variance score. We extend the original feature set of $\mathcal{M}_0$ with variance features added in and list the updated set below:

- translation probability expectation features in two directions: $\bar{p}_E(\bar{e}|\bar{f})$ and $\bar{p}_E(\bar{f}|\bar{e})$
- translation probability variance features in two directions: $p_V(\bar{e}|\bar{f})$ and $p_V(\bar{f}|\bar{e})$
- lexical weight expectation features in two directions: $\bar{p}_{E\,lex}(\bar{e}|\bar{f})$ and $\bar{p}_{E\,lex}(\bar{f}|\bar{e})$
- lexical weight variance features in two directions: $p_{V\,lex}(\bar{e}|\bar{f})$ and $p_{V\,lex}(\bar{f}|\bar{e})$

4 Experiments

4.1 Data Condition

We conduct experiments on the NIST Chinese-to-English MT tasks. We tune model parameters on the NIST 2003 (*MT03*) evaluation set by MERT (Och, 2003), and report results on NIST evaluation sets including the NIST 2004 (*MT04*), the NIST 2005 (*MT05*), the newswire portion of the NIST 2006 (*MT06*) and 2008 (*MT08*). Performances are measured in terms of the case-insensitive BLEU scores in percentage numbers. Table 1 gives statistics over these evaluation sets.

	MT03	MT04	MT05	MT06	MT08
Sent	919	1,788	1,082	616	691
Word	23,788	48,215	29,263	17,316	17,424

Table 1. Statistics on dev/test evaluation sets

We use the *selected data* that picked out from the whole data available for the NIST 2008 constrained track of Chinese-to-English machine translation task as the training corpora, including LDC2003E07, LDC2003E14, LDC2005T06, LDC2005T10, LDC2005E83, LDC2006E26, LDC2006E34, LDC2006E85 and LDC2006E92, which contain about 498,000 sentence pairs after pre-processing. Word alignments are performed by GIZA++ (Och and Ney, 2000) in both directions with an *intersect-diag-grow* refinement.

A traditional 5-gram language model (LM) for all involved systems is trained on the English side of all bilingual data plus the Xinhua portion of LDC English Gigaword Version 3.0. A lexicalized reordering model (Xiong *et al.*, 2006) is trained on the selected data in maximum entropy principle for the phrase-based system. A trigram target dependency LM (DLM) is trained on the English side of the selected data for the dependency-based hierarchical system.

4.2 MT System Description

We include four baseline systems. The first one (***Phr***) is a phrasal system (Xiong *et al.*, 2006) based on Bracketing Transduction Grammar (Wu, 1997) with a lexicalized reordering component based on maximum entropy model. The second one (***Hier***) is a hierarchical phrase-based system (Chiang, 2007) based on Synchronous Context Free Grammar (SCFG). The third one (***Dep***) is a string-to-dependency hierarchical phrase-based system (Shen *et al.*, 2008) with a dependency language model, which translates source strings to target dependency trees. The fourth one (***Synx***) is a syntax-based system (Galley *et al.*, 2006) that translates source strings to target syntactic trees.

4.3 TMG based on Multiple Paradigms

We develop TMG for each baseline system's TM based on the other three TMs as auxiliary models. *All prior probabilities of TMs are set equally to 0.25 heuristically as their similar performances.* Evaluation results are shown in Table 2, where gains more than 0.2 BLEU points are highlighted as improved cases. Compared to baseline systems, systems based on generalized TMs improve in most cases (18 times out of 20). We also notice that the improvements achieved on tree-based systems (Dep and Synx) are relatively smaller than those on string-based systems (Phr and Hier). A potential explanation can be that with considering more syntactic restrictions, tree-based systems suffer less than string-based systems on the over-estimation problem. We do not present further results with variance features added because of their consistent un-promising numbers. *We think this may be due to the considerable portion of non-overlapping translation pairs between main model and auxiliary models, which cause the variances not so accurate*.

		MT03(dev)	MT04	MT05	MT06	MT08	Average
Phr	Baseline	40.45	39.21	38.03	34.24	30.21	36.43
	TMG	41.19**(+0.74)**	39.74**(+0.53)**	38.39**(+0.36)**	34.71**(+0.47)**	30.69**(+0.48)**	36.94**(+0.51)**
Hier	Baseline	41.30	39.63	38.83	34.63	30.46	36.97
	TMG	41.67**(+0.37)**	40.25**(+0.62)**	39.11**(+0.28)**	35.78**(+1.15)**	31.17**(+0.71)**	37.60**(+0.63)**
Dep	Baseline	41.10	39.81	39.47	35.72	30.50	37.32
	TMG	41.37**(+0.27)**	39.92(+0.11)	39.91**(+0.44)**	35.99**(+0.27)**	31.07**(+0.57)**	37.65**(+0.33)**
Synx	Baseline	41.02	39.88	39.47	36.41	32.15	37.79
	TMG	41.26**(+0.24)**	40.09**(+0.21)**	39.90**(+0.43)**	36.77**(+0.36)**	32.15(+0.00)	38.03**(+0.24)**

Table 2. Results of TMG based on TMs with different paradigms

4.4 TMG based on Single Paradigm

We then evaluate TMG based on auxiliary models generated by the random sampling method.

We first decide the percentage of training data to be sampled. We empirically vary this number by 20%, 40%, 60%, 80% and 90% and use each sampled data to train an auxiliary model. We then run TMG on the baseline TM with different auxiliary model used each time. For time saving, we only evaluate on MT03 for Phr in Figure 2.

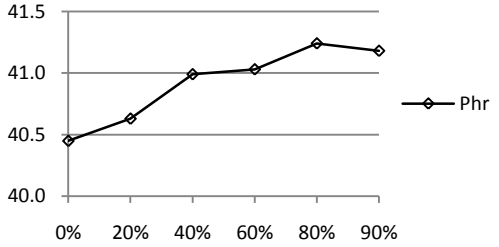

Figure 2. Affects of different percentages of data

The optimal result is achieved when the percentage is 80%, and we fix it as the default value in following experiments.

We then decide the number of auxiliary models used for TMG by varying it from 1 to 5. We list different results on MT03 for Phr in Figure 3.

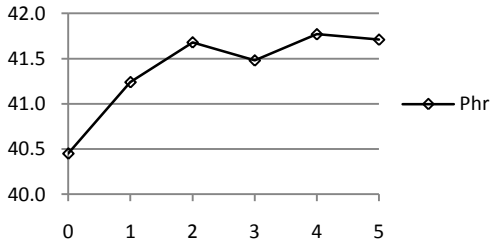

Figure 3. Affects of different numbers of auxiliary models

The optimal result is achieved when the number of auxiliary models is 4, and we fix it as the default value in following experiments.

We now develop TMG for each baseline system's TM based on auxiliary models constructed under default settings determined above. Evaluation results are shown in Table 3. We also investigate the affect of variance features for performance, whose results are denoted as ***TMG+Var***.

From Table 3 we can see that, compared to the results on baseline systems, systems using generalized TMs obtain improvements on almost all evaluation sets (19 times out of 20). With probability variance features added further, the improvements become even more stable than the ones using TMG only (20 times out of 20). Similar to the trend in Table 2, we also notice that TMG method is more preferred by string-based systems (Phr and Hier) rather than tree-based systems (Dep and Synx). This makes our conclusion more solidly that syntactic restrictions can help to alleviate the over-estimation problem.

4.5 Analysis on Phrase Coverage

We next empirically investigate on the translation pair coverage between TM ensembles built by different ways, and use them to analyze results got from previous experiments. Here, we only focus on *full lexicalized* translation entries between models. Those entries with variables are out of consideration in comparisons because of their model dependent properties.

Phrase pairs in the first three TMs have a length limitation in source side up to 3 words, and each source phrase can be translated to at most 20 target phrases.

		MT03(dev)	MT04	MT05	MT06	MT08	Average
Phr	Baseline	40.45	39.21	38.03	34.24	30.21	36.43
	TMG	41.77**(+1.32)**	40.28**(+1.07)**	39.13**(+1.10)**	35.38**(+1.14)**	31.12**(+0.91)**	37.54**(+1.11)**
	TMG+Var	41.77**(+1.32)**	40.31**(+1.10)**	39.43**(+1.30)**	35.61**(+1.37)**	31.62**(+1.41)**	37.74**(+1.31)**
Hier	Baseline	41.30	39.63	38.83	34.63	30.46	36.97
	TMG	42.28**(+0.98)**	40.45**(+0.82)**	39.61**(+0.78)**	35.67**(+1.04)**	31.54**(+1.08)**	37.91**(+0.94)**
	TMG+Var	42.42**(+1.12)**	40.55**(+0.92)**	39.69**(+0.86)**	35.55**(+0.92)**	31.41**(+0.95)**	37.92**(+0.95)**
Dep	Baseline	41.10	39.81	39.47	35.72	30.50	37.32
	TMG	41.49**(+0.39)**	40.20**(+0.39)**	40.00**(+0.53)**	36.13**(+0.41)**	31.24**(+0.74)**	37.81**(+0.49)**
	TMG+Var	41.72**(+0.62)**	40.57**(+0.76)**	40.44**(+0.97)**	36.15**(+0.43)**	31.31**(+0.81)**	38.04**(+0.72)**
Synx	Baseline	41.02	39.88	39.47	36.41	32.15	37.79
	TMG	41.18(+0.16)	40.30**(+0.42)**	39.90**(+0.43)**	36.99**(+0.58)**	32.45**(+0.30)**	38.16**(+0.37)**
	TMG+Var	41.42**(+0.40)**	40.55**(+0.67)**	40.17**(+0.70)**	36.89**(+0.48)**	32.51**(+0.36)**	38.31**(+0.52)**

Table 3. Results of TMG based on TMs constructed by random data sampling

For the fourth TM, these two limitations are released to 4 words and 30 target phrases. We treat phrase pairs identical on both sides but with different syntactic labels in the fourth TM as a unique pair for conveniences in statistics.

We first make statistics on TMs with different paradigms in Table 4. We can see from Table 4 that only slightly over half of the phrase pairs contained by the four involved TMs are common, which is also similar to the conclusion drawn in DeNeefe *et al.* (2006).

Models	#Translation Pair	#Percentage
Phr	1,222,909	**50.6%**
Hier	1,222,909	**50.6%**
Dep	1,087,198	**56.9%**
Synx	1,188,408	**52.0%**
Overlaps	618,371	-

Table 4. Rule statistics on TMs constructed by different paradigms

We then make statistics on TMs with identical paradigm in Table 5. For each baseline TM and its corresponding four auxiliary models constructed by random data sampling, we count the number of phrase pairs that are common between them and compute the percentage numbers based on it for each TM individually.

Models	TM_0	TM_1	TM_2	TM_3	TM_4
Phr	**61.8%**	74.0%	74.1%	73.9%	74.1%
Hier	**61.8%**	74.0%	74.1%	73.9%	74.1%
Dep	**60.8%**	73.6%	73.6%	73.5%	73.7%
Synx	**57.2%**	68.4%	68.5%	68.5%	68.6%

Table 5. Rule statistics on TMs constructed by random sampling (TM_0 is the main model)

Compared to the numbers in Table 4, we find that the coverage between baseline TM and sampled auxiliary models with identical paradigm is larger than that between baseline TM and auxiliary models with different paradigms (about 10 percents). It is a potential reason can explain why results of TMG based on sampled auxiliary models are more effective than those based on auxiliary models built with different paradigms, as we infer that *they share more common phrase pairs each other and make the computation of feature expectations and variances to be more reliable and accurate*.

4.6 Improvements on System Combination

Besides working for single-system decoding, we also perform a system combination method on *N*-best outputs from systems using generalized TMs. We re-implement a state-of-the-art word-level System Combination (**SC**) approach based on incremental HMM alignment proposed by Li *et al.* (2009a). The default number of *N*-best candidates used is set to 20.

We evaluate SC on *N*-best outputs generated from 4 baseline decoders by using different TM settings and list results in Table 6, where ***Base*** stands for combination results on systems using default TMs; ***Paras*** stands for combination results on systems using TMs generalized based on auxiliary models with different paradigms; and ***Samp*** stands for combination results on systems using TMs generalized based on auxiliary models constructed by the random data sampling method. For the Samp setting, we also include probability variance features computed based on Equation 3 in the log-linear model.

SC	MT03	MT04	MT05	MT06	MT08
Base	44.20	42.30	41.22	37.77	33.07
Paras	**44.40**	**42.69**	**41.53**	**38.05**	**33.31**
Samp	**44.80**	**42.95**	**42.10**	**38.39**	**33.67**

Table 6. Results on system combination

From Table 6 we can see that system combination can benefit from TMG method.

4.7 Improvements on Model Combination

As an alternative, model combination is another effective way to improve translation performance by utilizing multiple systems. We re-implement the Model Combination (**MC**) approach (DeNero *et al.*, 2010) using *N*-best lists as its inputs and develop it on *N*-best outputs used in Table 6. Evaluation results are presented in Table 7.

MC	MT03	MT04	MT05	MT06	MT08
Base	42.31	40.57	40.31	38.65	33.88
Paras	**42.87**	**40.96**	**40.77**	**38.81**	**34.47**
Samp	**43.29**	**41.29**	**41.11**	**39.28**	**34.77**

Table 7. Results on model combination

From Table 7 we can see that model combination can also benefit from TMG method.

5 Related Work

Foster and Kuhn (2007) presented an approach that resembles more to our work, in which they divided the training corpus into different components and integrated models trained on each component using the mixture modeling. However, their motivation was to address the *domain adaption problem*, and additional genre information should be provided for the corpus partition to create multiple models for mixture. We instead present two ways for the model ensemble construction without extra information needed: building models by different paradigms or by a random data sampling technique inspired by a machine learning technique. Compared to the prior work, our approach is more general, which can also be used for model adaptation. We can also treat TMG as a smoothing way to address the over-estimation problem existing in almost all TMs. Some literatures have paid attention to this issue as well, such as Foster *et al.* (2006) and Mylonakis and Sima 'an (2008). However, they did not leverage information between multiple models as we did, and developed on single models only. Furthermore, we also make current translation probability features to contain more statistical meanings by introducing the probability variance features into the log-linear model, which are completely novel to prior work and provide further improvements.

6 Conclusion and Future Work

In this paper, we have investigated a simple but effective translation model generalization method that benefits by integrating values of probability features between multiple TMs and using them in decoding phase directly. We also introduce novel probability variance features into the current feature sets of translation models and make the SMT models to be more flexible. We evaluate our method on four state-of-the-art SMT systems, and get promising results not only on single-system decodings, but also on a system combination approach and a model combination approach.

Making use of different distributions of translation probability features is the essential of this work. In the future, we will extend TMG method to other statistical models in SMT framework, (e.g. LM), which could be also suffered from the over-estimation problem. And we will make further research on how to tune prior probabilities of models automatically as well, in order to make our method to be more robust and tunable.

References

Auli Michael, Adam Lopez, Hieu Hoang, and Philipp Koehn. 2009. *A Systematic Analysis of Translation Model Search Spaces*. In *4th Workshop on Statistical Machine Translation*, pages 224-232.

Breiman Leo. 1996. *Bagging Predictors*. *Machine Learning*.

Chiang David. 2007. *Hierarchical Phrase Based Translation*. *Computational Linguistics*, 33(2): 201-228.

DeNero John, Shankar Kumar, Ciprian Chelba, and Franz Och. 2010. *Model Combination for Machine Translation.* To appear in *Proc. of the North American Chapter of the Association for Computational Linguistic.*

DeNeefe Steve, Kevin Knight, Wei Wang, and Daniel Marcu. 2007. *What Can Syntax-based MT Learn from Phrase-based MT?* In *Proc. of Empirical Methods on Natural Language Processing*, pages 755-763.

Foster George, Roland Kuhn, and Howard Johnson. 2006. *Phrasetable Smoothing for Statistical Machine Translation.* In *Proc. of Empirical Methods on Natural Language Processing*, pages 53-61.

Foster George and Roland Kuhn. 2007. *Mixture-Model Adaptation for SMT.* In *2th Workshop on Statistical Machine Translation*, pages 128-135.

Galley Michel, Jonathan Graehl, Kevin Knight, Daniel Marcu, Steve DeNeefe, Wei Wang, and Ignacio Thayer. 2006. *Scalable Inference and Training of Context-Rich Syntactic Translation Models*. In *Proc. of 44th Meeting of the Association for Computational Linguistics*, pages: 961-968.

Huang Liang. 2008. *Forest Reranking: Discriminative Parsing with Non-Local Features*. In *Proc. of 46th Meeting of the Association for Computational Linguistics*, pages 586-594.

Hoeting Jennifer, David Madigan, Adrian Raftery, and Chris Volinsky. 1999. *Bayesian Model Averaging: A tutorial. Statistical Science*, Vol. 14, pages 382-417.

He Xiaodong, Mei Yang, Jianfeng Gao, Patrick Nguyen, and Robert Moore. 2008. *Indirect-HMM-based Hypothesis Alignment for Combining Outputs from Machine Translation Systems*. In *Proc. of Empirical Methods on Natural Language Processing*, pages 98-107.

Koehn Philipp. 2004. *Phrase-based Model for SMT*. *Computational Linguistics*, 28(1): 114-133.

Li Chi-Ho, Xiaodong He, Yupeng Liu, and Ning Xi. 2009a. *Incremental HMM Alignment for MT system Combination*. In *Proc. of 47th Meeting of the Association for Computational Linguistics*, pages 949-957.

Li Mu, Nan Duan, Dongdong Zhang, Chi-Ho Li, and Ming Zhou. 2009b. *Collaborative Decoding: Partial Hypothesis Re-Ranking Using Translation Consensus between Decoders*. In *Proc. of 47th Meeting of the Association for Computational Linguistics*, pages 585-592.

Liu Yang, Haitao Mi, Yang Feng, and Qun Liu. 2009. *Joint Decoding with Multiple Translation Models*. In *Proc. of 47th Meeting of the Association for Computational Linguistics*, pages 576-584.

Mylonakis Markos and Khalil Sima 'an. 2008. *Phrase Translation Probabilities with ITG Priors and Smoothing as Learning Objective*. In *Proc. of Empirical Methods on Natural Language Processing*, pages 630-639.

Matusov Evgeny, Nicola Ueffi ng, and Hermann Ney. 2006. Computing consensus translation from multiple machine translation systems using enhanced hypotheses alignment. In *Proc. of European Charter of the Association for Computational Linguistics*, pages 33-40.

Och Franz and Hermann Ney. 2000. *Improved Statistical Alignment Models*. In *Proc. of 38th Meeting of the Association for Computational Linguistics*, pages 440-447.

Och Franz. 2003. *Minimum Error Rate Training in Statistical Machine Translation*. In *Proc. of 41th Meeting of the Association for Computational Linguistics*, pages 160-167.

Och Franz and Hermann Ney. 2004. *The Alignment template approach to Statistical Machine Translation*. *Computational Linguistics*, 30(4): 417-449.

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

Wu Dekai. 1997. *Stochastic Inversion Transduction Grammars and Bilingual Parsing of Parallel Corpora*. *Computational Linguistics*, 23(3): 377-404.

Xiong Deyi, Qun Liu, and Shouxun Lin. 2006. *Maximum Entropy based Phrase Reordering Model for Statistical Machine Translation*. In *Proc. of 44th Meeting of the Association for Computational Linguistics*, pages 521-528.

Zollmann Andreas, Ashish Venugopal, Franz Och, and Jay Ponte. 2008. *A Systematic Comparison of Phrase-Based, Hierarchical and Syntax-Augmented Statistical MT*. In *23rd International Conference on Computational Linguistics*, pages 1145-1152.