The Feature Subspace Method for SMT System Combination

Nan Duan[1], Mu Li[2], Tong Xiao[3], Ming Zhou[2]
[1]Tianjin University, Tianjin, China
[2]Microsoft Research Asia, Beijing, China
[3]Northeastern University, Shenyang, China
{v-naduan,muli,v-toxiao,mingzhou}@microsoft.com

Abstract

Recently system combination has been shown to be an effective way to improve translation quality over single machine translation systems. In this paper, we present a simple and effective method to systematically derive an ensemble of SMT systems from one baseline linear SMT model for use in system combination. Each system in the resulting ensemble is based on a feature set derived from the features of the baseline model (typically a subset of it). We will discuss the principles to determine the feature sets for derived systems, and present in detail the system combination model used in our work. Evaluation is performed on the data sets for NIST 2004 and NIST 2005 Chinese-to-English machine translation tasks. Experimental results show that our method can bring significant improvements to baseline systems with state-of-the-art performance.

1 Introduction

Research on Statistical Machine Translation (SMT) has shown substantial progress in recent years. Since the success of phrase-based methods (Och and Ney, 2004; Koehn, 2004), models based on formal syntax (Chiang, 2005) or linguistic syntax (Liu et al., 2006; Marcu et al., 2006) have also achieved state-of-the-art performance. As a result of the increasing numbers of available machine translation systems, studies on system combination have been drawing more and more attention in SMT research.

There have been many successful attempts to combine outputs from multiple machine translation systems to further improve translation quality. A system combination model usually takes *n*-best translations of single systems as input, and depending on the combination strategy, different methods can be used. Sentence-level combination methods directly select hypotheses from original outputs of single SMT systems (Sim et al., 2007; Hildebrand and Vogel, 2008), while phrase-level or word–level combination methods are more complicated and could produce new translations different from any translations in the input (Bangalore et al., 2001; Jayaraman and Lavie, 2005; Matusov et al., 2006; Sim et al., 2007).

Among all the factors contributing to the success of system combination, there is no doubt that the availability of multiple machine translation systems is an indispensable premise. Although various approaches to SMT system combination have been explored, including enhanced combination model structure (Rosti et al., 2007), better word alignment between translations (Ayan et al., 2008; He et al., 2008) and improved confusion network construction (Rosti et al., 2008), most previous work simply used the ensemble of SMT systems based on different models and paradigms at hand and did not tackle the issue of how to obtain the ensemble in a principled way. To our knowledge the only work discussed this problem is Macherey and Och (2007), in which they experimented with building different SMT systems by varying one or more sub-models (i.e. translation model or distortion model) of an existing SMT system, and observed that changes in early-stage model training introduced most diversities in translation outputs.

In this paper, we address the problem of building an ensemble of diversified machine translation systems from a single translation engine for system combination. In particular, we propose a novel *Feature Subspace* method for the ensemble construction based on any baseline SMT model which can be formulated as a standard linear function. Each system within the ensemble is based on a group of features directly derived from the baseline model with minimal efforts (which is typically a subset of the features used in the baseline model), and the resulting system is optimized in the derived feature space accordingly.

We evaluated our method on the test sets for NIST 2004 and NIST 2005 Chinese-to-English

Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing, pages 1096–1104,
Singapore, 6-7 August 2009.

machine translation tasks using two baseline SMT systems with state-of-the-art performance. Experimental results show that the feature subspace method can bring significant improvements to both baseline systems.

The rest of the paper is organized as follows. The motivation of our work is described on Section 2. In Section 3, we first give a detailed description about feature subspace method, including the principle to select subspaces from all possible options, and then an *n*-gram consensus – based sentence-level system combination method is presented. Experimental results are given in Section 4. Section 5 discusses some related issues and concludes the paper.

2 Motivation

Our motivations for this work can be described in the following two aspects.

The first aspect is related to the cost of building single systems for system combination. In previous work, the SMT systems used in combination differ mostly in two ways. One is the underlying models adopted by individual systems. For example, using an ensemble of systems respectively based on phrase-based models, hierarchical models or even syntax-based models is a common practice. The other is the methods used for feature function estimation such as using different word alignment models, language models or distortion models. For the first solution, building a new SMT system with different methodology is by no means an easy task even for an experienced SMT researcher, because it requires not only considerable effects to develop but also plenty of time to accumulate enough experiences to fine tune the system. For the second alternative, usually it requires time-consuming retraining for word alignment or language models. Also some of the feature tweaking in this solution is system or language specific, thus for any new systems or language pairs, human engineering has to be involved. For example, using different word segmentation methods for Chinese can generate different word alignment results, and based on which a new SMT system can be built. Although this may be useful to combination of Chinese-to-English translation, it is not applicable to most of other language pairs. Therefore it will be very helpful if there is a light-weight method that enables the SMT system ensemble to be systematically constructed based on an existing SMT system.

Source sentence	中国 最大 规模 的 海水 淡化 工程 落户 舟山
Ref translation	China's largest sea water desalinization project settles in Zhoushan
Default translation	China 's largest desalination project in Zhoushan
FS_{-PEF} translation	China 's largest sea water desalination project in Zhoushan

Table 1: An example of translations generated from the same decoder but with different feature settings.

	Chinese	English	$p(e\|f)$
1	海水 淡化	desalination	0.4000
2	海水	sea water	0.1748
3	淡化	desalination	0.0923

Table 2: Parameters of related phrases for examples in Table 1.

The second aspect motivating our work comes from the subspace learning method in machine learning literature (Ho, 1998), in which an ensemble of classifiers are trained on subspaces of the full feature space, and final classification results are based on the vote of all classifiers in the ensemble. Lopez and Resnik (2006) also showed that feature engineering could be used to overcome deficiencies of poor alignment. To illustrate the usefulness of feature subspace in the SMT task, we start with the example shown in Table 1. In the example, the Chinese source sentence is translated with two settings of a hierarchical phrase-based system (Chiang, 2005). In the default setting all the features are used as usual in the decoder, and we find that the translation of the Chinese word 海水 (sea water) is missing in the output. This can be explained with the data shown in Table 2. Because of noises and word alignment errors in the parallel training data, the inaccurate translation phrase 海水 淡化 ⇒ $desalination$ is assigned with a high value of the phrase translation probability feature $p(e|f)$. Although the correct translation can also be composed by two phrases 海水 ⇒ $sea\ water$ and 淡化 ⇒ $desalination$, its overall translation score cannot beat the incorrect one because the combined phrase translation probability of these two phrases are much smaller than $p(desalination|$海水 淡化$)$. However, if we intentionally remove the $p(e|f)$ feature from the model, the preferred translation can be generated as shown in the result of FS_{-PEF} because in

this way the bad estimation of $p(e|f)$ for this phrase is avoided.

This example gives us the hint that building decoders based on subspaces of a standard model could help with working around some negative impacts of inaccurate estimations of feature values for some input sentences. The subspace-based systems are expected to work similarly to statistical classifiers trained on subspaces of a full feature space – though the overall accuracy of baseline system might be better than any individual systems, for a specific sentence some individual systems could generate better translations. It is expected that employing an ensemble of subspace-based systems and making use of consensus between them will outperform the baseline system.

3 Feature Subspace Method for SMT System Ensemble Construction

In this section, we will present in detail the method for systematically deriving SMT systems from a standard linear SMT model based on feature subspaces for system combination.

3.1 SMT System Ensemble Generation

Nowadays most of the state-of-the-art SMT systems are based on linear models as proposed in Och and Ney (2002). Let $h_m(f,e)$ be a feature function, and λ_m be its weight, an SMT model D can be formally written as:

$$e^* = \arg\max_e \sum_m \lambda_m h_m(f,e) \quad (1)$$

Noticing that Equation (1) is a general formulation independent of any specific features, technically for any subset of features used in D, a new SMT system can be constructed based on it, which we call a *sub-system*.

Next we will use Ω to denote the full feature space defined by the entire set of features used in D, and $s \subseteq \Omega$ is a feature subset that belongs to $\rho(\Omega)$, the power set of Ω. The derived sub-system based on subset $s \subseteq \Omega$ is denoted by d_s. Although in theory we can use all the sub-systems derived from every feature subset in $\rho(\Omega)$, it is still desirable to use only some of them in practice. The reasons for this are two-fold. First, the number of possible sub-systems ($2^{|\Omega|}$) is exponential to the size of Ω. Even when the number of features in Ω is relatively small, i.e. 10, there will be up to 1024 sub-systems in total, which is a large number for combination task. Larger feature sets will make the system combination practically infeasible. Second, not every sub-system could contribute to the system combination. For example, feature subsets only containing very small number of features will lead to sub-systems with very poor performance; and the language model feature is too important to be ignored for a sub-system to achieve reasonably good performance.

In our work, we only consider feature subspaces with only one difference from the features in Ω. For each non- language model feature h_i, a sub-system d_i is built by removing h_i from Ω. Allowing for the importance of the language model (LM) feature to an SMT model, we do not remove any LM feature from any sub-system. Instead, we try to weaken the strength of a LM feature by lowering its *n*-gram order. For example, if a 4-gram language model is used in the baseline system D, then a trigram model can be used in one sub-system, and a bigram model can be used in another. In this way more than one sub-system can be derived based on one LM feature. When varying a language model feature, the *one-feature difference* principle is still kept: if we lower the order of a language model feature, no other features are removed or changed.

The remaining issue of using weakened LM features is that the resulting ensemble is no longer strictly based on subspace of Ω. However, this theoretical imperfection can be remedied by introducing Ω', a super-space of Ω to include all lower-order LM features. In this way, an augmented baseline system D' can be built based on Ω', and the baseline system D itself can also be viewed as a sub-system of D'. We will show in the experimental section that D' actually performs even slightly better than the original baseline system D, but results of sub-system combination are significantly better that both D and D'.

After the sub-system ensemble is constructed, each sub-system tunes its feature weights independently to optimize the evaluation metrics on the development set.

Let $\mathcal{D} = \{d_1, \ldots, d_n\}$ be the set of sub-systems obtained by either removing one non-LM feature or changing the order of a LM feature, and $\mathcal{H}_i$ be the *n*-best list produced by d_i. Then $\mathcal{H}(\mathcal{D})$, the *translation candidate pool* to the system combination model can be written as:

$$\mathcal{H}(\mathcal{D}) = \bigcup_i \mathcal{H}_i \quad (2)$$

The advantage of this method is that it allows us to systematically build an ensemble of SMT systems at a very low cost. From the decoding

perspective, all the sub-systems share a common decoder, with minimal extensions to the baseline systems to support the use of specified subset of feature functions to compute the overall score for translation hypotheses. From the model training perspective, all the non-LM feature functions can be estimated once for all sub-systems. The only exception is the language model feature, which may be of different values across multiple sub-systems. However, since lower-order models have already been contained in higher-order model for the purpose of smoothing in almost all statistical language model implementations, there is also no extra training cost.

3.2 System Combination Scheme

In our work, we use a sentence-level system combination model to select best translation hypothesis from the candidate pool $\mathcal{H}(\mathcal{D})$. This method can also be viewed to be a hypotheses re-ranking model since we only use the existing translations instead of performing decoding over a confusion network as done in the word-level combination method (Rosti et al., 2007).

The score function in our combination model is formulated as follows:

$$e^* = \underset{e \in \mathcal{H}(\mathcal{D})}{argmax} \lambda_{LM} h_{LM}(e) + \lambda_l L + \psi(e, \mathcal{H}(\mathcal{D})) \tag{3}$$

where $h_{LM}(e)$ is the language model score for e, L is the length of e, and $\psi(e, \mathcal{H}(\mathcal{D}))$ is a translation consensus –based scoring function. The computation of $\psi(e, \mathcal{H}(\mathcal{D}))$ is further decomposed into weighted linear combination of a set of *n*-gram consensus –based features, which are defined in terms of the order of *n*-gram to be matched between current candidate and other translation in $\mathcal{H}(\mathcal{D})$.

Given a translation candidate e, the *n*-gram agreement feature between e and other translations in the candidate pool is defined as:

$$h_n^+(e, \mathcal{H}(\mathcal{D})) = \sum_{e' \in \mathcal{H}(\mathcal{D}), e' \neq e} G_n(e, e') \tag{4}$$

where the function $G_n(e, e')$ counts the occurrences of *n*-grams of e in e':

$$G_n(e, e') = \sum_{i=1}^{|e|-n+1} \delta(e_i^{i+n-1}, e') \tag{5}$$

Here $\delta(\cdot,\cdot)$ is the indicator function - $\delta(e_i^{i+n-1}, e')$ is 1 when the *n*-gram e_i^{i+n-1} appears in e', otherwise it is 0.

In order to give the combination model an opportunity to penalize long but inaccurate translations, we also introduce a set of *n*-gram disagreement features in the combination model:

$$h_n^-(e, \mathcal{H}(\mathcal{D})) = \sum_{e' \in \mathcal{H}(\mathcal{D}), e' \neq e} (|e| - n + 1 - G_n(e, e')) \tag{6}$$

Because each order of *n*-gram match introduces two features, the total number of features in the combination model will be $2m + 2$ if m orders of *n*-gram are to be matched in computing $\psi(e, \mathcal{H}(\mathcal{D}))$. Since we also adopt a linear scoring function in Equation (3), the feature weights of our combination model can also be tuned on a development data set to optimize the specified evaluation metrics using the standard Minimum Error Rate Training (MERT) algorithm (Och 2003).

Our method is similar to the work proposed by Hildebrand and Vogel (2008). However, except the language model and translation length, we only use intra-hypothesis *n*-gram agreement features as Hildebrand and Vogel did and use additional intra-hypothesis *n*-gram disagreement features as Li et al. (2009) did in their co-decoding method.

4 Experiments

4.1 Data

Experiments were conducted on the NIST evaluation sets of 2004 (MT04) and 2005 (MT05) for Chinese-to-English translation tasks. Both corpora provide 4 reference translations per source sentence. Parameters were tuned with MERT algorithm (Och, 2003) on the NIST evaluation set of 2003 (MT03) for both the baseline systems and the system combination model. Translation performance was measured in terms of case-insensitive NIST version of BLEU score which computes the brevity penalty using the shortest reference translation for each segment, and all the results will be reported in percentage numbers. Statistical significance is computed using the bootstrap re-sampling method proposed by Koehn (2004). Statistics of the data sets are summarized in Table 3.

Data set	#Sentences	#Words
MT03 (dev)	919	23,782
MT04 (test)	1,788	47,762
MT05 (test)	1,082	29,258

Table 3: Data set statistics.

We use the parallel data available for the NIST 2008 constrained track of Chinese-to-English machine translation task as bilingual training data, which contains 5.1M sentence pairs, 128M Chinese words and 147M English words after pre-processing. GIZA++ toolkit (Och and Ney, 2003) is used to perform word alignment in both directions with default settings, and the intersect-diag-grow method is used to generate symmetric word alignment refinement. The language model used for all systems is a 5-gram model trained with the English part of bilingual data and Xinhua portion of LDC English Gigaword corpus version 3. In experiments, multiple language model features with the order ranging from 2 to 5 can be easily obtained from the 5-gram one without retraining.

4.2 System Description

Theoretically our method is applicable to all linear model –based SMT systems. In our experiments, two in-house developed systems are used to validate our method. The first one (**SYS1**) is a system based on the hierarchical phrase-based model as proposed in (Chiang, 2005). Phrasal rules are extracted from all bilingual sentence pairs, while hierarchical rules with variables are extracted from selected data sets including LDC2003E14, LDC2003E07, LDC2005T06 and LDC2005T10, which contain around 350,000 sentence pairs, 8.8M Chinese words and 10.3M English words. The second one (**SYS2**) is a re-implementation of a phrase-based decoder with lexicalized reordering model based on maximum entropy principle proposed by Xiong et al. (2006). All bilingual data are used to extract phrases up to length 3 on the source side.

In following experiments, we only consider removing common features shared by both baseline systems for feature subspace generation. Rule penalty feature and lexicalized reordering feature, which are particular to SYS1 and SYS2, are not used. We list the features in consideration as follows:

- ***PEF*** and ***PFE***: phrase translation probabilities $p(e|f)$ and $p(f|e)$
- ***PEFLEX*** and ***PFELEX***: lexical weights $p_{lex}(e|f)$ and $p_{lex}(f|e)$
- ***PP***: phrase penalty
- ***WP***: word penalty
- ***BLP***: bi-lexicon pair counting how many entries of a conventional lexicon co-occurring in a given translation pair
- ***LM-n***: language model with order *n*

Based on the principle described in Section 3.1, we generate a number of feature subspaces for each baseline system as follows:

- *For non-LM features* (*PEF, PFE, PEFLEX, PFELEX, PP*, *WP* and *BLP*), we remove one of them from the full feature space each time. Thus 7 feature subspaces are generated, which are denoted as FS_{-PEF}, FS_{-PFE}, $FS_{-PEFLEX}$, $FS_{-PFELEX}$, FS_{-PP}, FS_{-WP} and FS_{-BLP} respectively. The 5-gram LM feature is used in each of them.
- *For LM features* (*LM-n*), we change the order from 2 to 5 with all the other non-LM features present. Thus 4 LM-related feature subspaces are generated, which are denoted as FS_{LM-2}, FS_{LM-3}, FS_{LM-4} and FS_{LM-5} respectively. FS_{LM-5} is essentially the full feature space of baseline system.

For each baseline system, we construct a total of 11 sub-systems by using above feature subspaces. The baseline system is also contained within them because of using FS_{LM-5}. We call all sub-systems are *non-baseline sub-systems* except the one derived by using FS_{LM-5}.

By default, the beam size of 60 is used for all systems in our experiments. The size of *n*-best list is set to 20 for each sub-system, and for baseline systems, this size is set to 220, which equals to the size of the combined *n*-best list generated by total 11 sub-systems. The order of *n*-gram agreement and disagreement features used in sentence-level combination model ranges from unigram to 4-gram.

4.3 Evaluation of Oracle Translations

We first evaluate the oracle performance on the *n*-best lists of baseline systems and on the combined *n*-best lists of sub-systems generated from each baseline system.

The oracle translations are obtained by using the metric of sentence-level BLEU score (Ye et al., 2007). Table 4 shows the evaluation results, in which ***Baseline*** stands for baseline system with a 5-gram LM feature, and ***FS*** stands for 11 sub-systems derived from the baseline system.

		SYS1	SYS2
		BLEU/TER	BLEU/TER
MT04	*Baseline*	49.68/0.6411	49.50/0.6349
	FS	**51.05/0.6089**	**50.53/0.6056**
MT05	*Baseline*	48.89/0.5946	48.37/0.5944
	FS	**50.69/0.5695**	**49.81/0.5684**

Table 4: Oracle BLEU and TER scores on baseline systems and their generated sub-systems.

For both SYS1 and SYS2, feature subspace method achieves higher oracle BLEU and lower TER scores on both MT04 and MT05 test sets, which gives the feature subspace method more potential to achieve higher performance than the baseline systems.

We then investigate the ratio of translation candidates in the combined *n*-best lists of non-baseline sub-systems that are not included in the baseline's *n*-best list. Table 5 shows the statistics.

	MT04	MT05
SYS1	69.71%	69.69%
SYS2	59.07%	58.54%

Table 5: Ratio of unique translation candidates from non-baseline sub-systems.

From Table 5 we can see that only less than half of the translation candidates of sub-systems overlap with those the of baseline systems. This result, together with the oracle BLEU and TER score estimation, helps eliminate the concern that no diversities or better translation candidates can be obtained by using sub-systems.

4.4 Feature Subspace Method on Single SMT System

Next we validate the effect of feature subspace method on single SMT systems.

Figure 1 shows the evaluation results of different systems on the MT05 test set. From the figure we can see that the overall accuracy of baseline systems is better than any of their derived sub-systems, and except the sub-system derived by using FS_{LM-2}, the performance of all the systems are fairly similar.

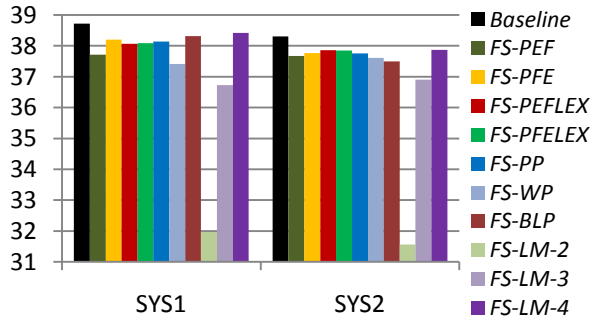

Figure 1: Performances of different systems.

We then evaluate the system combination method proposed in Section 3.2 with all the sub-systems for each baseline system. Table 6 shows the results on both MT04 and MT05 data sets, in which ***FS-Comb*** denotes the system combination using 11 sub-systems.

From Table 6 we can see that by using *FS-Comb* we obtain about 1.1~1.3 points of BLEU gains over baseline systems. We also include in Table 6 the results for ***Baseline+mLM***, which stands for the augmented baseline system as described in Section 3.1 using a bunch of LM features from bigram to 5-gram. It can be seen that both augmented baseline systems outperform their corresponding baseline systems slightly but consistently on both data sets.

		MT04	MT05
SYS1	*Baseline*	39.07	38.72
	Baseline+mLM	39.34+	39.14+
	FS-Comb	**40.43**++	**39.79**++
SYS2	*Baseline*	38.84	38.30
	Baseline+mLM	38.95*	38.63+
	FS-Comb	**39.92**++	**39.49**++

Table 6: Translation results of *Baseline*, *Baseline+mLM* and *FS-Comb* (+: significant better than baseline system with $p < 0.05$; ++: significant better than baseline system with $p < 0.01$; *: no significant improvement).

We also investigate the results when we incrementally add the *n*-best list of each sub-system into a candidate pool to see the effects when different numbers of sub-systems are used in combination. In order to decide the sequence of sub-systems to add, we first evaluate the performance of pair-wise combinations between each sub-system and its baseline system on the development set. That is, for each sub-system, we combine its *n*-best list with the *n*-best list of its baseline system and perform system combination for MT03 data set. Then we rank the sub-systems by the pair-wise combination performance from high to low, and use this ranking as the sequence to add *n*-best lists of sub-systems. Each time when a new *n*-best list is added, the combination performance based on the enlarged candidate pool is evaluated. Figure 2 shows the results on both MT04 and MT05 test sets, in which ***SYS1-fs*** and ***SYS2-fs*** denote the sub-systems derived from SYS1 and SYS2 respectively, and X-axis is the number of sub-systems used for combination each time and Y-axis is the BLEU score. From the figure we can see that although in some cases the performance slightly drops when a new sub-system is added, generally using more sub-systems always leads to better results.

Next we examine the performance of baseline systems when different beam sizes are used in decoding. The results on MT05 test set are shown in Figure3, where X-axis is the beam size. In Figure 3, ***SYS1+mLM*** and ***SYS2+mLM*** denote augmented baseline systems of SYS1 and SYS2 with multiple LM features.

From Figure 3 we can see that augmented baseline systems (with multiple LM features) outperform the baseline systems (with only one LM feature) for all beam sizes ranging from 20 to 220. In this experiment we did not observe any significant performance improvements when using larger beam sizes than the default setting, but using more sub-systems in combination almost always bring improvements.

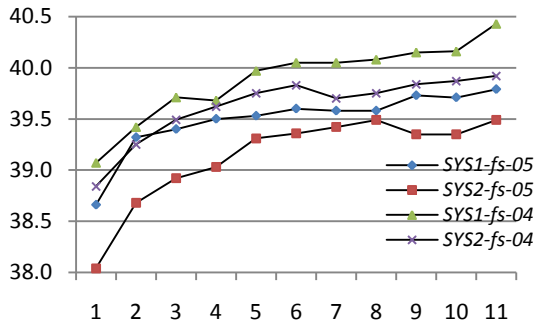

Figure 2: Performances on different numbers of sub-systems.

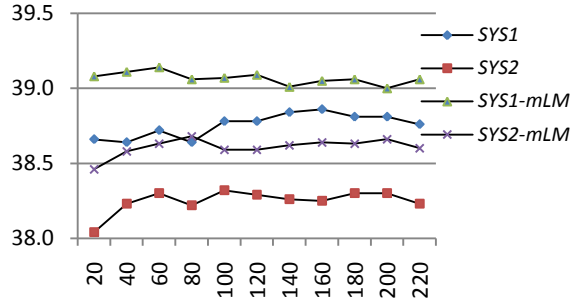

Figure 3: Performances on different beam sizes.

	MT04	MT05
SYS1-fs	44.63%	46.12%
SYS2-fs	47.54%	44.73%

Table 7: Ratio of final translations coming from non-baseline sub-systems.

Finally, we investigate the ratio of final translations coming from the *n*-best lists of non-baseline sub-systems only. Table 7 shows the results on both MT04 and MT05 test sets, which indicate that almost half of the final translations are contributed by the non-baseline sub-systems.

4.5 The Impact of *n*-best List Size

In order to find the optimal size of *n*-best list for combination, we compare the combination results of using list sizes from 10-best up to 500-best for each sub-system.

In this experiment, system combination was performed on the combined *n*-best list from total 11 sub-systems with different list size each time. Figure 4 shows the results on the MT03 dev set and the MT04 and MT05 test sets for both SYS1 and SYS2. X-axis is the *n*-best list size of each sub-system.

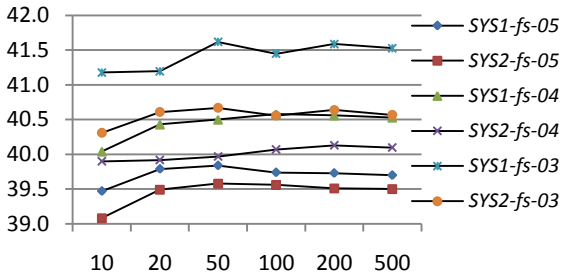

Figure 4: Performances on different *n*-best sizes.

We can see from the figure that for all data sets the optimal *n*-best list size is around 50, but the improvements are not significant over the results when 20-best translations are used. The reason for the small optimal *n*-best list size could be that the low-rank hypotheses might introduce more noises into the combined translation candidate pool for sentence-level combination (Hasan et al., 2007; Hildebrand and Vogel, 2008).

4.6 Feature Subspace Method on Multiple SMT Systems

In the last experiment, we investigate the effect of feature subspace method when multiple SMT systems are used in system combination.

Evaluation results are reported in Table 8. The system combination method described in Section 3.2 is used to combine outputs from two baseline systems (with only one 5-gram LM feature) and sub-systems generated from both baseline systems (22 in total), with their results denoted as ***Baseline Comb* (*both*)** and ***FS Comb* (*both*)** respectively. We also include the combination results of sub-systems based on one baseline system for reference in the table.

On both MT04 and MT05 test sets, the results of system combination based on sub-systems are significantly better than those of baseline systems, which show that our method can also help with system combination when more than one system are used. We can also see that using multiple systems based on different SMT models and using our subspace based method can help each other: the best performance can only be achieved when both are employed.

	MT04	MT05
Baseline Comb (*both*)	39.98	39.43
FS-Comb (SYS1)	40.43	39.79
FS-Comb (SYS2)	39.92	39.49
FS Comb (*both*)	**40.96**	**40.38**

Table 8: Performances of sentence-level combination on multiple SMT systems.

5 Conclusion

In this paper, we have presented a novel and effective *Feature Subspace* method for the construction of an ensemble of machine translation systems based on a baseline SMT model which can be formulated as a standard linear function. Each system within the ensemble is based on a subset of features derived from the baseline model, and the resulting ensemble can be used in system combination to improve translation quality. Experimental results on NIST Chinese-to-English translation tasks show that our method can bring significant improvements to two baseline systems with state-of-the-art performance, and it is expected that our method can be employed to improve any linear model -based SMT systems. There is still much room for improvements in the current work. For example, we still use a simple one-feature difference principle for feature subspace generation. In the future, we will explore more possibilities for feature subspaces selection and experiment with our method in a word-level system combination model.

References

Necip Fazil Ayan, Jing Zheng, and Wen Wang. 2008. Improving alignments for better confusion networks for combining machine translation systems. In *Proc. COLING*, pages 33-40.

Srinivas Bangalore, German Bordel, and Giuseppe Riccardi. 2001. Computing consensus translation from multiple machine translation systems. In *Proc. ASRU*, pages 351-354.

David Chiang. 2005. A hierarchical phrase-based model for statistical machine translation. In *Proc. ACL*, pages 263-270.

Xiaodong He, Mei Yang, Jianfeng Gao, Patrick Nguyen, and Robert Moore. 2008. Indirect-hmm-based hypothesis for combining outputs from machine translation systems. In *Proc. EMNLP*, pages 98-107.

Almut Silja Hildebrand and Stephan Vogel. 2008. Combination of machine translation systems via hypothesis selection from combined n-best lists. In *8th AMTA conference*, pages 254-261.

Tin Kam Ho. 1998. The random subspace method for constructing decision forests. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 832-844.

Sasa Hasan, Richard Zens, and Hermann Ney. 2007. Are very large *n*-best lists useful for SMT? In *Proc. NAACL, Short paper*, pages 57-60.

S. Jayaraman and A. Lavie. 2005. Multi-Engine Machine Translation Guided by Explicit Word Matching. In *10th EAMT conference*, pages 143-152.

Philipp Koehn. 2004. Statistical significance tests for machine translation evaluation. In *Proc. EMNLP*, pages 388-395.

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

Mu Li, Nan Duan, Dongdong Zhang, Chi-Ho Li, and Ming Zhou. 2009. Collaborative Decoding: Partial Hypothesis Re-Ranking Using Translation Consensus between Decoders. In *Proc. ACL-IJCNLP*.

Adam Lopez and Philip Resnik. 2006. Word-Based Alignment, Phrase-Based Translation: What's the link? In *7th AMTA conference*, pages 90-99.

Yang Liu, Qun Liu, and Shouxun Lin. 2006. Tree-to-String Alignment Template for Statistical Machine Translation. In *Proc. ACL*, pages 609-616.

Daniel Marcu, Wei Wang, Abdessamad Echihabi, and Kevin Knight. 2006. SPMT: Statistical machine translation with syntactified target language phrases. In *Proc. EMNLP*, pages 44-52.

Wolfgang Macherey and Franz Och. 2007. An Empirical Study on Computing Consensus Translations from Multiple Machine Translation Systems. In *Proc. EMNLP*, pages 986-995.

Evgeny Matusov, Nicola Ueffi ng, and Hermann Ney. 2006. Computing consensus translation from multiple machine translation systems using enhanced hypotheses alignment. In *Proc. EACL*, pages 33-40.

Franz Och and Hermann Ney. 2002. Discriminative training and maximum entropy models for statis-

tical machine translation. In *Proc. ACL*, pages 295-302.

Franz Och. 2003. Minimum error rate training in statistical machine translation. In *Proc. ACL*, pages 160-167.

Franz Och and Hermann Ney. 2003. A systematic comparison of various statistical alignment models. *Computational Linguistics*, 29(1): pages 19-51.

Franz Och and Hermann Ney. 2004. The alignment template approach to statistical machine translation. *Computational Linguistics*, 30(4): pages 417-449.

Antti-Veikko Rosti, Necip Fazil Ayan, Bing Xiang, Spyros Matsoukas, Richard Schwartz, and Bonnie Dorr. 2007. Combining outputs from multiple machine translation systems. In *Proc. NAACL*, pages 228-235.

Antti-Veikko Rosti, Spyros Matsoukas, and Richard Schwartz. 2007. Improved Word-Level System Combination for Machine Translation. In *Proc. ACL*, pages 312-319.

Antti-Veikko Rosti, Bing Zhang, Spyros Matsoukas, and Richard Schwartz. 2008. Incremental hypothesis alignment for building confusion networks with application to machine translation system combination. In *Proc. Of the Third ACL Workshop on Statistical Machine Translation*, pages 183-186.

K.C. Sim, W. Byrne, M. Gales, H. Sahbi, and P. Woodland. 2007. Consensus network decoding for statistical machine translation system combination. In *ICASSP*, pages 105-108.

Deyi Xiong, Qun Liu, and Shouxun Lin. 2006. Maximum entropy based phrase reordering model for statistical machine translation. In *Proc. ACL*, pages 521-528.

Yang Ye, Ming Zhou, and Chin-Yew Lin. 2007. Sentence level Machine Translation Evaluation as a Ranking Problem: One step aside from BLEU. In *Proc. Of the Second ACL Workshop on Statistical Machine Translation*, pages 240-247.