Collaborative Decoding: Partial Hypothesis Re-ranking Using Translation Consensus between Decoders

Mu Li[1], Nan Duan[2], Dongdong Zhang[1], Chi-Ho Li[1], Ming Zhou[1]
[1]Microsoft Research Asia
Beijing, China
[2]Tianjin University
Tianjin, China
{muli,v-naduan,dozhang,chl,mingzhou}@microsoft.com

Abstract

This paper presents collaborative decoding (co-decoding), a new method to improve machine translation accuracy by leveraging translation consensus between multiple machine translation decoders. Different from system combination and MBR decoding, which post-process the n-best lists or word lattice of machine translation decoders, in our method multiple machine translation decoders collaborate by exchanging partial translation results. Using an iterative decoding approach, n-gram agreement statistics between translations of multiple decoders are employed to re-rank both full and partial hypothesis explored in decoding. Experimental results on data sets for NIST Chinese-to-English machine translation task show that the co-decoding method can bring significant improvements to all baseline decoders, and the outputs from co-decoding can be used to further improve the result of system combination.

1 Introduction

Recent research has shown substantial improvements can be achieved by utilizing consensus statistics obtained from outputs of multiple machine translation systems. Translation consensus can be measured either at sentence level or at word level. For example, Minimum Bayes Risk (MBR) (Kumar and Byrne, 2004) decoding over *n*-best list tries to find a hypothesis with lowest expected loss with respect to all the other translations, which can be viewed as sentence-level consensus-based decoding. Word based methods proposed range from straightforward consensus voting (Bangalore et al., 2001; Matusov et al., 2006) to more complicated word-based system combination model (Rosti et al., 2007; Sim et al., 2007). Typically, the resulting systems take outputs of individual machine translation systems as input, and build a new confusion network for second-pass decoding.

There have been many efforts dedicated to advance the state-of-the-art performance by combining multiple systems' outputs. Most of the work focused on seeking better word alignment for consensus-based confusion network decoding (Matusov et al., 2006) or word-level system combination (He et al., 2008; Ayan et al., 2008). In addition to better alignment, Rosti et al. (2008) introduced an incremental strategy for confusion network construction; and Hildebrand and Vogel (2008) proposed a hypotheses re-ranking model for multiple systems' outputs with more features including word translation probability and *n*-gram agreement statistics.

A common property of all the work mentioned above is that the combination models work on the basis of *n*-best translation lists (full hypotheses) of existing machine translation systems. However, the *n*-best list only presents a very small portion of the entire search space of a Statistical Machine Translation (SMT) model while a majority of the space, within which there are many potentially good translations, is pruned away in decoding. In fact, due to the limitations of present-day computational resources, a considerable number of promising possibilities have to be abandoned at the early stage of the decoding process. It is therefore expected that exploring additional possibilities beyond *n*-best hypotheses lists for full sentences could bring improvements to consensus-based decoding.

In this paper, we present *collaborative decoding* (or *co-decoding*), a new SMT decoding scheme to leverage consensus information between multiple machine translation systems. In this scheme, instead of using a post-processing step, multiple machine translation decoders collaborate during the decoding process, and translation consensus statistics are taken into account to improve ranking not only for full translations, but also for partial hypotheses. In this way, we expect to reduce search errors caused by partial hypotheses pruning, maximize the contribution of translation consensus, and result in better final translations.

We will discuss the general co-decoding model, requirements for decoders that enable collaborative decoding and describe the updated model structures. We will present experimental results on the data sets of NIST Chinese-to-English machine translation task, and demonstrate that co-decoding can bring significant improvements to baseline systems. We also conduct extensive investigations when different settings of co-decoding are applied, and make comparisons with related methods such as word-level system combination of hypothesis selection from multiple *n*-best lists.

The rest of the paper is structured as follows. Section 2 gives a formal description of the co-decoding model, the strategy to apply consensus information and hypotheses ranking in decoding. In Section 3, we make detailed comparison between co-decoding and related work such as system combination and hypotheses selection out of multiple systems. Experimental results and discussions are presented in Section 4. Section 5 concludes the paper.

2 Collaborative Decoding

2.1 Overview

Collaborative decoding does not present a full SMT model as other SMT decoders do such as Pharaoh (Koehn, 2004) or Hiero (Chiang, 2005). Instead, it provides a framework that accommodates and coordinates multiple MT decoders. Conceptually, collaborative decoding incorporates the following four constituents:

1. *Co-decoding model.* A co-decoding model consists of a set of *member models*, which are a set of augmented *baseline models*. We call decoders based on member models *member decoders*, and those based on baseline models *baseline decoders*. In our work, any Maximum A Posteriori (MAP) SMT model with log-linear formulation (Och, 2002) can be a qualified candidate for a baseline model. The requirement for a log-linear model aims to provide a natural way to integrate the new co-decoding features.
2. *Co-decoding features.* Member models are built by adding additional translation consensus -based co-decoding features to baseline models. A baseline model can be viewed as a special case of member model with all co-decoding feature values set to 0. Accordingly, a baseline decoder can be viewed as a special setting of a member decoder.
3. *Decoder coordinating*. In co-decoding, each member decoder cannot proceed solely based on its own agenda. To share consensus statistics with others, the decoding must be performed in a coordinated way.
4. *Model training.* Since we use multiple inter-related decoders and introduce more features in member models, we also need to address the parameter estimation issue in the framework of co-decoding.

In the following sub-sections we first establish a general model for co-decoding, and then present details of feature design and decoder implementation, as well as parameter estimation in the co-decoding framework. We leave the investigation of using specific member models to the experiment section.

2.2 Generic Collaborative Decoding Model

For a given source sentence *f*, a member model in co-decoding finds the best translation e^* among the set of possible candidate translations $\mathcal{H}(f)$ based on a scoring function F:

$$e^* = \text{argmax}_{e \in \mathcal{H}(f)} F(e) \quad (1)$$

In the following, we will use d_k to denote the k^{th} member decoder, and also use the notation $\mathcal{H}_k(f)$ for the translation hypothesis space of f determined by d_k. The m^{th} member model can be written as:

$$F_m(e) = \Phi_m(f, e) + \sum\nolimits_{k,k \neq m} \Psi_k(e, \mathcal{H}_k(f)) \quad (2)$$

where $\Phi_m(f, e)$ is the score function of the m^{th} baseline model, and each $\Psi_k(e, \mathcal{H}_k(f))$ is a partial consensus score function with respect to d_k and is defined over e and $\mathcal{H}_k(f)$:

$$\Psi_k(e, \mathcal{H}_k(f)) = \sum\nolimits_l \lambda_{k,l}\, h_{k,l}(e, \mathcal{H}_k(f)) \quad (3)$$

where each $h_{k,l}(e, \mathcal{H}_k(f))$ is a feature function based on a consensus measure between e and $\mathcal{H}_k(f)$, and $\lambda_{k,l}$ is the corresponding feature weight. Feature index l ranges over all consensus-based features in Equation 3.

2.3 Decoder Coordination

Before discussing the design and computation of translation consensus -based features, we first describe the multiple decoder coordination issue in co-decoding. Note that in Equation 2, though the baseline score function $\Phi_m(f,e)$ can be computed inside each decoder, the case of $\Psi_k(e,\mathcal{H}_k(f))$ is more complicated. Because usually it is not feasible to enumerate the entire hypothesis space for machine translation, we approximate $\mathcal{H}_k(f)$ with *n*-best hypotheses by convention. Then there is a circular dependency between co-decoding features and $\mathcal{H}_k(f)$: on one hand, searching for *n*-best approximation of $\mathcal{H}_k(f)$ requires using Equation 2 to select top-ranked hypotheses; while on the other hand, Equation 2 cannot be computed until every $\mathcal{H}_k(f)$ is available.

We address this issue by employing a bootstrapping method, in which the key idea is that we can use baseline models' *n*-best hypotheses as seeds, and iteratively refine member models' *n*-best hypotheses with co-decoding. Similar to a typical phrase-based decoder (Koehn, 2004), we associate each hypothesis with a coverage vector *c* to track translated source words in it. We will use $\mathcal{H}_k(c,f)$ for the set of hypotheses associated with c, and we also denote with $\mathcal{H}_k(f) = \bigcup_c \mathcal{H}_k(c,f)$ the set of all hypotheses generated by member decoder d_k in decoding. The co-decoding process can be described as follows:

1. For each member decoder d_k, perform decoding with a baseline model, and memorize all translation hypotheses generated during decoding in $\mathcal{H}_k(f)$;
2. Re-group translation hypotheses in $\mathcal{H}_k(f)$ into a set of buckets $\mathcal{H}_k(c,f)$ by the coverage vector *c* associated with each hypothesis;
3. Use member decoders to re-decode source sentence f with member models. For member decoder d_k, consensus-based features of any hypotheses associated with coverage vector *c* are computed based on current setting of $\mathcal{H}_s(c,f)$ for all *s* but *k*. New hypotheses generated by d_k in re-decoding are cached in $\mathcal{H}_k^{'}(f)$;
4. Update all $\mathcal{H}_k(f)$ with $\mathcal{H}_k^{'}(f)$;
5. Iterate from step 2 to step 4 until a preset iteration limit is reached.

In the iterative decoding procedure described above, hypotheses of different decoders can be mutually improved. For example, given two decoders d_1 and d_2 with hypotheses sets $\mathcal{H}_1$ and $\mathcal{H}_2$, improvements on $\mathcal{H}_1$ enable d_2 to improve $\mathcal{H}_2$, and in turn $\mathcal{H}_1$ benefits from improved $\mathcal{H}_2$, and so forth.

Step 2 is used to facilitate the computation of feature functions $h_{k,l}(e,\mathcal{H}_k(\cdot))$, which require both e and every hypothesis in $\mathcal{H}_k(\cdot)$ should be translations of the same set of source words. This step seems to be redundant for CKY-style MT decoders (Liu et al., 2006; Xiong et al., 2006; Chiang, 2005) since the grouping is immediately available from decoders because all hypotheses spanning the same range of source sentence have been stacked together in the same chart cell. But to be a general framework, this step is necessary for some state-of-the-art phrase-based decoders (Koehn, 2007; Och and Ney, 2004) because in these decoders, hypotheses with different coverage vectors can co-exist in the same bin, or hypotheses associated with the same coverage vector might appear in different bins.

Note that a member model does not enlarge the theoretical search space of its baseline model, the only change is hypothesis scoring. By re-running a complete decoding process, member model can be applied to re-score all hypotheses explored by a decoder. Therefore step 3 can be viewed as full-scale hypothesis re-ranking because the re-ranking scope is beyond the limited *n*-best hypotheses currently cached in $\mathcal{H}_k$.

In the implementation of member decoders, there are two major modifications compared to their baseline decoders. One is the support for co-decoding features, including computation of feature values and the use of augmented co-decoding score function (Equation 2) for hypothesis ranking and pruning. The other is hypothesis grouping based on coverage vector and a mechanism to effectively access grouped hypotheses in step 2 and step 3.

2.4 Co-decoding Features

We now present the consensus-based feature functions $h_{k,l}(e,\mathcal{H}_k(f))$ introduced in Equation 3. In this work all the consensus-based features have the following formulation:

$$h_{k,l}\big(e,\mathcal{H}_k(f)\big) = \sum_{e' \in \mathcal{H}_k(f)} P(e'|d_k) G_l(e,e') \quad (4)$$

where e is a translation of *f* by decoder $d_m(m \neq k)$, $e^{'}$ is a translation in $\mathcal{H}_k(f)$ and $P(e'|d_k)$ is the posterior probability of translation $e^{'}$ determined by decoder d_k given source sentence *f*. $G_l(e,e')$ is a consensus measure defined on *e* and e', by varying which different feature functions can be obtained.

Referring to the log-linear model formulation, the translation posterior $P(e'|d_k)$ can be computed as:

$$P(e'|d_k) = \frac{\exp\big(\alpha F_k(e')\big)}{\sum_{e'' \in \mathcal{H}_k(f)} \exp\big(\alpha F_k(e'')\big)} \qquad (5)$$

where $F_k(\cdot)$ is the score function given in Equation 2, and α is a scaling factor following the work of Tromble et al. (2008)

To compute the consensus measures, we further decompose each $G_l(e, e')$ into n-gram matching statistics between e and e'. Here we do not discriminate among different lexical n-grams and are only concerned with statistics aggregation of all n-grams of the same order. For each n-gram of order n, we introduce a pair of complementary consensus measure functions $G_{n^+}(e, e')$ and $G_{n^-}(e, e')$ described as follows:

$G_{n^+}(e, e')$ is the n-gram agreement measure function which counts the number of occurrences in e' of n-grams in e. So the corresponding feature value will be the expected number of occurrences in $\mathcal{H}_k(f)$ of all n-grams in e:

$$G_{n^+}(e, e') = \sum_{i=1}^{|e|-n+1} \tau(e_i^{i+n-1}, e')$$

where $\tau(\cdot,\cdot)$ is a binary indicator function – $\tau\big(e_i^{i+n-1}, e'\big)$ is 1 if the n-gram e_i^{i+n-1} occurs in e' and 0 otherwise.

$G_{n^-}(e, e')$ is the n-gram disagreement measure function which is complementary to $G_{n^+}(e, e')$:

$$G_{n^-}(e, e') = \sum_{i=1}^{|e|-n+1} \Big(1 - \tau\big(e_i^{i+n-1}, e'\big)\Big)$$

This feature is designed because $G_{n^+}(e, e')$ does not penalize long translation with low precision. Obviously we have the following:

$$G_{n^+}(e, e') + G_{n^-}(e, e') = |e| - n + 1$$

So if the weights of agreement and disagreement features are equal, the disagreement-based features will be equivalent to the translation length features. Using disagreement measures instead of translation length there could be two potential advantages: 1) a length feature has been included in the baseline model and we do not need to add one; 2) we can scale disagreement features independently and gain more modeling flexibility.

Similar to a language model score, n-gram consensus -based feature values cannot be summed up from smaller hypotheses. Instead, it must be re-computed when building each new hypothesis.

2.5 Model Training

We adapt the Minimum Error Rate Training (MERT) (Och, 2003) algorithm to estimate parameters for each member model in co-decoding. Let $\boldsymbol{\lambda}_m$ be the feature weight vector for member decoder d_m, the training procedure proceeds as follows:

1. Choose initial values for $\boldsymbol{\lambda}_1, \ldots, \boldsymbol{\lambda}_M$
2. Perform co-decoding using all member decoders on a development set D with $\boldsymbol{\lambda}_1, \ldots, \boldsymbol{\lambda}_M$. For each decoder d_m, find a new feature weight vector $\boldsymbol{\lambda}'_m$ which optimizes the specified evaluation criterion L on D using the MERT algorithm based on the n-best list $\mathcal{H}_m$ generated by d_m:

   $$\boldsymbol{\lambda}'_m = \mathrm{argmax}_{\boldsymbol{\lambda}}\, L\,(T|\boldsymbol{\lambda}, \mathcal{H}_m\,, D))$$

   where T denotes the translations selected by re-ranking the translations in $\mathcal{H}_m$ using a new feature weight vector $\boldsymbol{\lambda}$
3. Let $\boldsymbol{\lambda}_1 = \boldsymbol{\lambda}'_1, \ldots, \boldsymbol{\lambda}_M = \boldsymbol{\lambda}'_M$ and repeat step 2 until convergence or a preset iteration limit is reached.

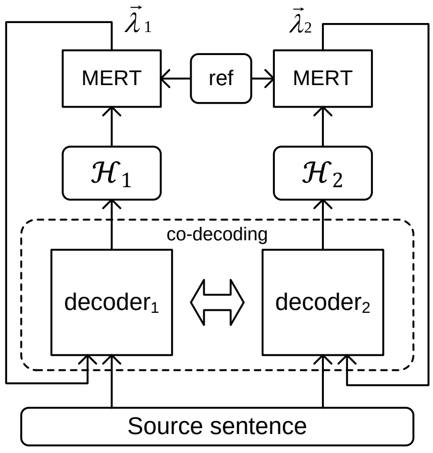

Figure 1. Model training for co-decoding

In step 2, there is no global criterion to optimize the co-decoding parameters across member models. Instead, parameters of different member models are tuned to maximize the evaluation criteria on each member decoder's own n-best output. Figure 1 illustrates the training process of co-decoding with 2 member decoders.

2.6 Output Selection

Since there is more than one model in co-decoding, we cannot rely on member model's score function to choose one best translation from multiple decoders' outputs because the model scores are not directly comparable. We will examine the following two system combination -based solutions to this task:

- Word-level system combination (Rosti et al., 2007) of member decoders' *n*-best outputs
- Hypothesis selection from combined *n*-best lists as proposed in Hildebrand and Vogel (2008)

3 Experiments

In this section we present experiments to evaluate the co-decoding method. We first describe the data sets and baseline systems.

3.1 Data and Metric

We conduct our experiments on the test data from the NIST 2005 and NIST 2008 Chinese-to-English machine translation tasks. The NIST 2003 test data is used for development data to estimate model parameters. Statistics of the data sets are shown in Table 1. In our experiments all the models are optimized with case-insensitive NIST version of BLEU score and we report results using this metric in percentage numbers.

Data set	# Sentences	# Words
NIST 2003 (dev)	919	23,782
NIST 2005 (test)	1,082	29,258
NIST 2008 (test)	1,357	31,592

Table 1: 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++ 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 models (include decoding models and system combination models described in Section 2.6) is a 5-gram model trained with the English part of bilingual data and xinhua portion of LDC English Gigaword corpus version 3.

3.2 Member Decoders

We use three baseline decoders in the experiments. The first one (SYS1) is re-implementation of Hiero, a hierarchical phrase-based decoder. Phrasal rules are extracted from all bilingual sentence pairs, while rules with variables are extracted only 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 BTG decoder with lexicalized reordering model based on maximum entropy principle as proposed by Xiong et al. (2006). We use all the bilingual data to extract phrases up to length 3. The third one (SYS3) is a string-to-dependency tree –based decoder as proposed by Shen et al. (2008). For rule extraction we use the same setting as in SYS1. We parsed the language model training data with Berkeley parser, and then trained a dependency language model based on the parsing output. All baseline decoders are extended with *n*-gram consensus –based co-decoding features to construct member decoders. By default, the beam size of 20 is used for all decoders in the experiments. We run two iterations of decoding for each member decoder, and hold the value of α in Equation 5 as a constant 0.05, which is tuned on the test data of NIST 2004 Chinese-to-English machine translation task.

3.3 Translation Results

We first present the overall results of co-decoding on both test sets using the settings as we described. For member decoders, up to 4-gram agreement and disagreement features are used. We also implemented the word-level system combination (Rosti et al., 2007) and the hypothesis selection method (Hildebrand and Vogel, 2008). 20-best translations from all decoders are used in the experiments for these two combination methods. Parameters for both system combination and hypothesis selection are also tuned on NIST 2003 test data. The results are shown in Table 2.

	NIST 2005	NIST 2008
SYS1	38.66/40.08	27.67/29.19
SYS2	38.04/39.93	27.25/29.14
SYS3	39.50/40.32	28.75/29.68
Word-level Comb	40.45/40.85	29.52/30.35
Hypo Selection	40.09/40.50	29.02/29.71

Table 2: Co-decoding results on test data

In the Table 2, the results of a member decoder and its corresponding baseline decoder are grouped together with the later one for the member decoders. On both test sets, every member decoder performs significantly better than its baseline decoder (using the method proposed in Koehn (2004) for statistical significance test).

We apply system combination methods to the *n*-best outputs of both baseline decoders and member decoders. We notice that we can achieve even better performance by applying system combination methods to member decoders' *n*-best outputs. However, the improvement margins are smaller than those of baseline decoders on both test sets. This could be the result of less diversified outputs from co-decoding than those from baseline decoders. In particular, the results for hypothesis selection are only slightly better than the best system in co-decoding.

We also evaluate the performance of system combination using different *n*-best sizes, and the results on NIST 2005 data set are shown in Figure 2, where *bl*- and *co*- legends denote combination results of baseline decoding and co-decoding respectively. From the results we can see that combination based on co-decoding's outputs performs consistently better than that based on baseline decoders' outputs for all *n*-best sizes we experimented with. However, we did not observe any significant improvements for both combination schemes when *n*-best size is larger than 20.

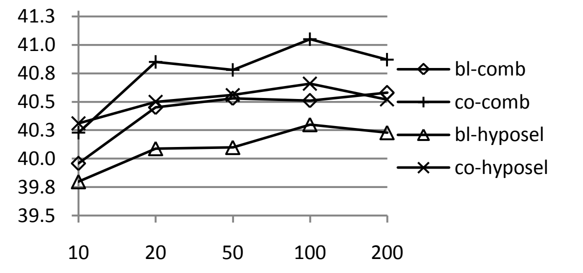

Figure 2. Performance of system combination with different sizes of *n*-best lists

One interesting observation in Table 2 is that the performance gap between baseline decoders is narrowed through co-decoding. For example, the 1.5 points gap between SYS2 and SYS3 on NIST 2008 data set is narrowed to 0.5. Actually we find that the TER score between two member decoders' outputs are significantly reduced (as shown in Table 3), which indicates that the outputs become more similar due to the use of consensus information. For example, the TER score between SYS2 and SYS3 of the NIST 2008 outputs are reduced from 0.4238 to 0.2665.

	NIST 2005	NIST 2008
SYS1 vs. SYS2	0.3190/0.2274	0.4016/0.2686
SYS1 vs. SYS3	0.3252/0.1840	0.4176/0.2469
SYS2 vs. SYS3	0.3498/0.2171	0.4238/0.2665

Table 3: TER scores between co-decoding translation outputs

In the rest of this section we run a series of experiments to investigate the impacts of different factors in co-decoding. All the results are reported on NIST 2005 test set.

We start with investigating the performance gain due to partial hypothesis re-ranking. Because Equation 3 is a general model that can be applied to both partial hypothesis and *n*-best (full hypothesis) re-scoring, we compare the results of both cases. Figure 3 shows the BLEU score curves with up to 1000 candidates used for re-ranking. In Figure 3, the suffix *p* denotes results for partial hypothesis re-ranking, and *f* for *n*-best re-ranking only. For partial hypothesis re-ranking, obtaining more top-ranked results requires increasing the beam size, which is not affordable for large numbers in experiments. We work around this issue by approximating beam sizes larger than 20 by only enlarging the beam size for the span covering the entire source sentence. From Figure 3 we can see that all decoders can gain improvements before the size of candidate set reaches 100. When the size is larger than 50, co-decoding performs consistently and significantly better than the re-ranking results on any baseline decoder's *n*-best outputs.

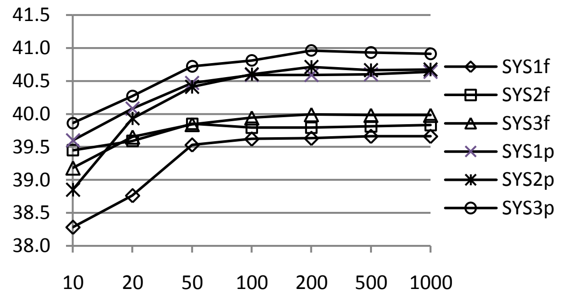

Figure 3. Partial hypothesis vs. *n*-best re-ranking results on NIST 2005 test data

Figure 4 shows the BLEU scores of a two-system co-decoding as a function of re-decoding iterations. From the results we can see that the results for both decoders converge after two iterations.

In Figure 4, iteration 0 denotes decoding with baseline model. The setting of iteration 1 can be viewed as the case of *partial co-decoding*, in which one decoder uses member model and the other keeps using baseline model. The results show that member models help each other: although improvements can be made using a single member model, best BLEU scores can only be achieved when both member models are used as shown by the results of iteration 2. The results also help justify the independent parameter estimation of member decoders described in Section 2.5, since optimizing the performance of one decoder will eventually bring performance improvements to all member decoders.

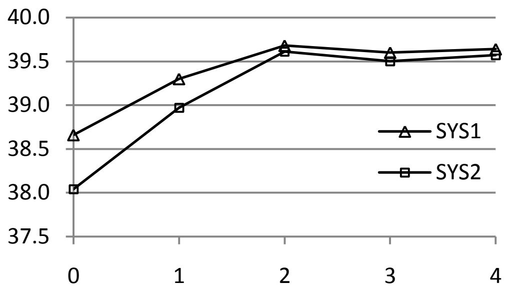

Figure 4. Results using incremental iterations in co-decoding

Next we examine the impacts of different consensus-based features in co-decoding. Table 4 shows the comparison results of a two-system co-decoding using different settings of *n*-gram agreement and disagreement features. It is clearly shown that both *n*-gram agreement and disagreement types of features are helpful, and using them together is the best choice.

	SYS1	SYS2
Baseline	38.66	38.04
+agreement –disagreement	39.36	39.02
–agreement +disagreement	39.12	38.67
+agreement +disagreement	39.68	39.61

Table 4: Co-decoding with/without *n*-gram agreement and disagreement features

In Table 5 we show in another dimension the impact of consensus-based features by restricting the maximum order of *n*-grams used to compute agreement statistics.

	SYS1	SYS2
1	38.75	38.27
2	39.21	39.10
3	39.48	39.25
4	39.68	39.61
5	39.52	39.36
6	39.58	39.47

Table 5: Co-decoding with varied *n*-gram agreement and disagreement features

From the results we do not observe BLEU improvement for $n > 4$. One reason could be that the data sparsity for high-order *n*-grams leads to over fitting on development data.

We also empirically investigated the impact of scaling factor α in Equation 5. It is observed in Figure 5 that the optimal value is between 0.01 ~ 0.1 on both development and test data.

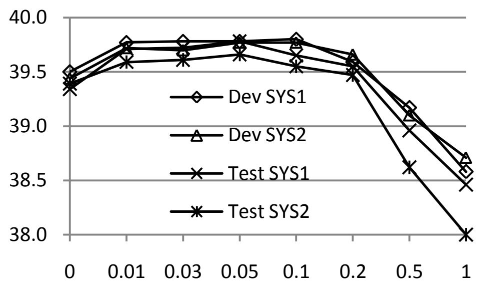

Figure 5. Impact of scaling factor α

4 Discussion

Word-level system combination (system combination hereafter) (Rosti et al., 2007; He et al., 2008) has been proven to be an effective way to improve machine translation quality by using outputs from multiple systems. Our method is different from system combination in several ways. System combination uses unigram consensus only and a standalone decoding model irrelevant to single decoders. Our method uses agreement information of *n*-grams, and consensus features are integrated into decoding models. By constructing a confusion network, system combination is able to generate new translations different from any one in the input *n*-best lists, while our method does not extend the search spaces of baseline decoding models. Member decoders only change the scoring and ranking of the candidates in the search spaces. Results in Table 2 show that these two approaches can be used together to obtain further improvements.

The work on multi-system hypothesis selection of Hildebrand and Vogel (2008) bears more resemblance to our method in that both make use of *n*-gram agreement statistics. They also empirically show that *n*-gram agreement is the most important factor for improvement apart from language models.

Lattice MBR decoding (Tromble et al., 2008) also uses *n*-gram agreement statistics. Their work focuses on exploring larger evidence space by using a translation lattice instead of the *n*-best list. They also show the connection between expected *n*-gram change and corpus Log-BLEU loss.

5 Conclusion

Improving machine translation with multiple systems has been a focus in recent SMT research. In this paper, we present a framework of collaborative decoding, in which multiple MT decoders are coordinated to search for better translations by re-ranking partial hypotheses using augmented log-linear models with translation consensus -based features. An iterative approach is proposed to re-rank all hypotheses explored in decoding. Experimental results show that with collaborative decoding every member decoder performs significantly better than its baseline decoder. In the future, we will extend our method to use lattice or hypergraph to compute consensus statistics instead of *n*-best lists.

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.

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

Philipp Koehn, 2004. Pharaoh: a beam search decoder for phrase-based statistical machine translation model. In *Proc. 6th AMTA Conference,* pages 115-124.

Philipp Koehn, Hieu Hoang, Alexandra Brich, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondrej Bojar, Alexandra Constantin, Evan Herbst. 2007. Moses: open source toolkit for statistical machine translation. In *Proc. ACL, demonstration session*.

Shankar Kumar and William Byrne 2004. Minimum Bayes-Risk Decoding for Statistical Machine Translation. In *HLT-NAACL*, pages 169-176.

Yang Liu, Qun Liu, Shouxun Lin. 2006. Tree-to-string alignment template for statistical machine translation. In *Proc. ACL-Coling*, pages 609-616.

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

Franz Och and Hermann Ney. 2002. Discriminative training and maximum entropy models for statistical 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. 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 *HLT-NAACL*, pages 228-235

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*.

Libin Shen, Jinxi Xu, and Ralph Weischedel. 2008. A new string-to-dependency machine translation algorithm with a target dependency language model. In *Proc. HLT-ACL*, pages 577-585.

Roy W. Tromble, Shankar Kumar, Franz Och, and Wolfgang Macherey. 2008. Lattice minimum bayes-risk decoding for statistical machine translation. In *Proc. EMNLP*, pages 620-629.

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