Hypothesis Mixture Decoding for Statistical Machine Translation

Nan Duan,
School of Computer Science and Technology
Tianjin University
Tianjin, China
v-naduan@microsoft.com

Mu Li, and **Ming Zhou**
Natural Language Computing Group
Microsoft Research Asia
Beijing, China
{muli,mingzhou}@microsoft.com

Abstract

This paper presents *hypothesis mixture decoding* (HM decoding), a new decoding scheme that performs translation reconstruction using hypotheses generated by multiple translation systems. HM decoding involves two decoding stages: first, each component system decodes independently, with the explored search space kept for use in the next step; second, a new search space is constructed by composing existing hypotheses produced by all component systems using a set of rules provided by the HM decoder itself, and a new set of model independent features are used to seek the final best translation from this new search space. Few assumptions are made by our approach about the underlying component systems, enabling us to leverage SMT models based on arbitrary paradigms. We compare our approach with several related techniques, and demonstrate significant BLEU improvements in large-scale Chinese-to-English translation tasks.

1 Introduction

Besides tremendous efforts on constructing more complicated and accurate models for statistical machine translation (SMT) (Och and Ney, 2004; Chiang, 2005; Galley et al., 2006; Shen et al., 2008; Chiang 2010), many researchers have concentrated on the approaches that improve translation quality using information between hypotheses from one or more SMT systems as well.

System combination is built on top of the *N*-best outputs generated by multiple component systems (Rosti et al., 2007; He et al., 2008; Li et al., 2009b) which aligns multiple hypotheses to build confusion networks as new search spaces, and outputs the highest scoring paths as the final translations. *Consensus decoding*, on the other hand, can be based on either single or multiple systems: single system based methods (Kumar and Byrne, 2004; Tromble et al., 2008; DeNero et al., 2009; Kumar et al., 2009) re-rank translations produced by a single SMT model using either *n*-gram posteriors or expected *n*-gram counts. Because hypotheses generated by a single model are highly correlated, improvements obtained are usually small; recently, dedicated efforts have been made to extend it from single system to multiple systems (Li et al., 2009a; DeNero et al., 2010; Duan et al., 2010). Such methods select translations by optimizing consensus models over the combined hypotheses using all component systems' posterior distributions.

Although these two types of approaches have shown consistent improvements over the standard Maximum a Posteriori (MAP) decoding scheme, most of them are implemented as post-processing procedures over translations generated by MAP decoders. In this sense, the work of Li et al. (2009a) is different in that both partial and full hypotheses are re-ranked during the decoding phase directly using consensus between translations from different SMT systems. However, their method does not change component systems' search spaces.

This paper presents *hypothesis mixture decoding* (HM decoding), a new decoding scheme that performs translation reconstruction using hypotheses generated by multiple component systems. HM decoding involves two decoding stages: first, each component system decodes the source sentence independently, with the explored search space kept for use in the next step; second, a new search space is constructed by composing existing hypo-

Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics, pages 1258–1267,
Portland, Oregon, June 19-24, 2011. ©2011 Association for Computational Linguistics

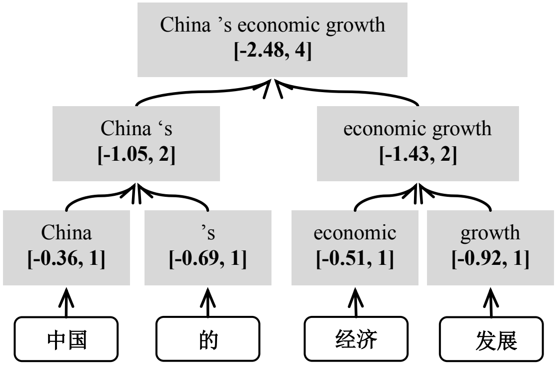

Figure 1: A decoding example of a phrase-based SMT system. Each hypothesis is annotated with a feature vector, which includes a logarithmic probability feature and a word count feature.

theses produced by all component systems using a set of rules provided by the HM decoder itself, and a new set of component model independent features are used to seek the final best translation from this new constructed search space.

We evaluate by combining two SMT models with state-of-the-art performances on the NIST Chinese-to-English translation tasks. Experimental results show that our approach outperforms the best component SMT system by up to 2.11 BLEU points. Consistent improvements can be observed over several related decoding techniques as well, including word-level system combination, collaborative decoding and model combination.

2 Hypothesis Mixture Decoding

2.1 Motivation and Overview

SMT models based on different paradigms have emerged in the last decade using fairly different levels of linguistic knowledge. Motivated by the success of system combination research, the key contribution of this work is to make more effective use of the extended search spaces from different SMT models in decoding phase directly, rather than just post-processing their final outputs. We first begin with a brief review of single system based SMT decoding, and then illustrate major challenges to this end.

Given a source sentence f, an SMT decoder seeks for a target translation e that best matches f as its translation by maximizing the following conditional probability:

$$P(e|f) = \frac{exp\,\{\sum_{r\in\mathcal{D}(e)} \theta \cdot \phi(r)\}}{\sum_{e'\in\mathcal{H}(f)} exp\,\{\sum_{r\in\mathcal{D}(e')} \theta \cdot \phi(r)\}}$$

where $\phi(\cdot)$ is the feature vector that includes a set of system specific features, θ is the weight vector, $\mathcal{D}(e)$ is a derivation that can yield e and is defined as a sequence of translation rule applications $\{r\}$. Figure 1 illustrates a decoding example, in which the final translation is generated by recursively composing partial hypotheses that cover different ranges of the source sentence until the whole input sentence is fully covered, and the feature vector of the final translation is the aggregation of feature vectors of all partial hypotheses used.[1]

However, hypotheses generated by different SMT systems cannot be combined directly to form new translations because of two major issues:

The first one is the heterogeneous structures of different SMT models. For example, a string-to-tree system cannot use hypotheses generated by a phrase-based system in decoding procedure, as such hypotheses are based on flat structures, which cannot provide any additional information needed in the syntactic model.

The second one is the incompatible feature spaces of different SMT models. For example, even if a phrase-based system can use the lexical forms of hypotheses generated by a syntax-based system without considering syntactic structures, the feature vectors of these hypotheses still cannot be aggregated together in any trivial way, because the feature sets of SMT models based on different paradigms are usually inconsistent.

To address these two issues discussed above, we propose HM decoding that performs translation reconstruction using hypotheses generated by multiple component systems.[2] Our method involves two decoding stages depicted as follows:

1. *Independent decoding* stage, in which each component system decodes input sentences independently based on its own model and search algorithm, and the explored search spaces (translation forests) are kept for use in the next stage.

[1] There are also features independent of translation derivations, such as the language model feature.

[2] In this paper, we will constrain our discussions within CKY-style decoders, in which we find translations for all spans of the source sentence. Although standard implementations of phrase-based decoders fall out of this scope, they can be still re-written to work in the CKY-style bottom-up manner at the cost of 1) only BTG-style reordering allowed, and 2) higher time complexity. As a result, any phrase-based SMT system can be used as a component in our HM decoding method.

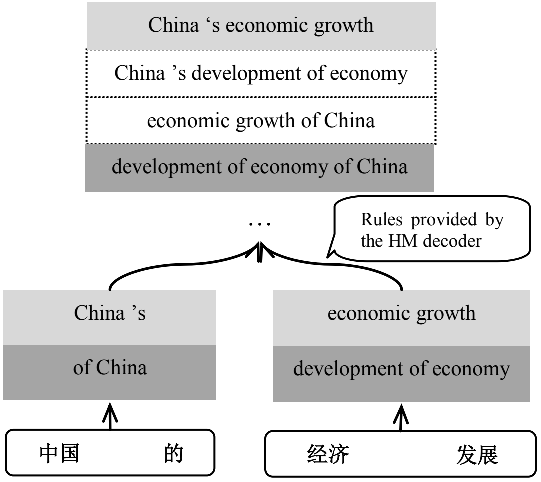

Figure 2: An example of HM decoding, in which the translations surrounded by the dotted lines are newly generated hypotheses. Hypotheses light-shaded come from a phrase-based system, and hypotheses dark-shaded come from a syntax-based system.

2. *HM decoding* stage, where a mixture search space is constructed for translation derivations by composing partial hypotheses generated by all component systems, and a new decoding model with a set of enriched feature functions are used to seek final translations from this newly generated search space.

HM decoding can use lexicalized hypotheses of arbitrary SMT models to derive translation, and a set of component model independent features are used to compute translation confidence. We discuss mixture search space construction, details of model and feature designs as well as HM decoding algorithms in Section 2.2, 2.3 and 2.4 respectively.

2.2 Mixture Search Space Construction

Let $\mathcal{M}_1, \dots, \mathcal{M}_N$ denote N component MT systems, f_i^j denote the span of a source sentence f starting at position i and ending at position j. We use $\mathcal{H}_n(f_i^j)$ denoting the search space of f_i^j predicted by $\mathcal{M}_n$, and $\mathcal{H}(f_i^j)$ denoting the mixture search space of f_i^j constructed by the HM decoder, which is defined recursively as follows:

- $\mathcal{H}_n(f_i^j) \subset \mathcal{H}(f_i^j)$. This rule adds all component systems' search spaces into the mixture search space for use in HM decoding. Thus hypotheses produced by all component systems are still available to the HM decoder.
- $r(e_{p_1}^{q_1}, \dots, e_{p_k}^{q_k}) \subset \mathcal{H}(f_i^j)$, in which $i \le p_k \le q_k \le j$ and $e_{p_k}^{q_k} \subset \mathcal{H}(f_{p_k}^{q_k})$. r is a translation rule provided by HM decoder that composes a new hypothesis using smaller hypotheses in the search spaces $\mathcal{H}(f_{p_1}^{q_1}), \dots, \mathcal{H}(f_{p_k}^{q_k})$. These rules further extend $\mathcal{H}(f_i^j)$ with hypotheses generated by the HM decoder itself.

Figure 2 shows an example of HM decoding, in which hypotheses generated by two SMT systems are used together to compose new translations. Since search space pruning is the indispensable procedure for all SMT systems, we will omit its explicit expression in the following descriptions and algorithms for convenience.

2.3 Models and Features

Following the common practice in SMT research, we use a linear model to formulate the preference of translation hypotheses in the mixture search space $\mathcal{H}(f)$. Formally, we are to find a translation $\hat{e}$ that maximizes the weighted linear combination of a set of real-valued features as follows:

$$\hat{e} = \underset{e \in \mathcal{H}(f)}{argmax} \left\{ \sum_i \lambda_i \cdot h_i(e,f) \right\}$$

where $h_i(e,f)$ is an HM decoding feature with its corresponding feature weight λ_i.

In this paper, the HM decoder does not assume the availability of any internal knowledge of the underlying component systems. The HM decoding features are independent of component models as well, which fall into two categories:

The first category contains a set of consensus-based features, which are inspired by the success of consensus decoding approaches. These features are described in details as follows:

1) $h_{\mathcal{H}_n(f)}(e,f)$: the *n*-gram posterior feature of e computed based on the component search space $\mathcal{H}_n(f)$ generated by $\mathcal{M}_n$:

$$h_{\mathcal{H}_n(f)}(e,f) = \sum_{\omega \in e} \#_{\omega}(e) p(\omega|\mathcal{H}_n(f))$$

$p(\omega|\mathcal{H}_n(f)) = \sum_{e' \in \mathcal{H}_n(f)} \delta_{\omega}(e') P_n(e'|f)$ is the posterior probability of an *n*-gram ω in $\mathcal{H}_n(f)$, $\#_{\omega}(e)$ is the number of times that ω occurs in e, $\delta_{\omega}(e)$ equals to 1 when ω occurs in e, and 0 otherwise.

2) $h_{\mathcal{H}_n^S(f)}(e^S, f)$: the stemmed *n*-gram posterior feature of e computed based on the stemmed component search space $\mathcal{H}_n^S(f)$. A word stem dictionary that includes 22,660 entries is used to convert e and $\mathcal{H}_n(f)$ into their stem forms e^S and $\mathcal{H}_n^S(f)$ by replacing each word into its stem form. This feature is computed similarly to that of $h_{\mathcal{H}_n(f)}(e, f)$.

3) $h_{\mathcal{H}(f)}(e, f)$: the *n*-gram posterior feature of e computed based on the mixture search space $\mathcal{H}(f)$ generated by the HM decoder:

$$h_{\mathcal{H}(f)}(e, f) = \sum_{\omega \in e} \#_{\omega}(e) p(\omega | \mathcal{H}(f))$$

$p(\omega | \mathcal{H}(f)) = \sum_{e' \in \mathcal{H}(f)} \delta_{\omega}(e') P(e' | f)$ is the posterior probability of an *n*-gram ω in $\mathcal{H}(f)$, $P(e' | f)$ is the posterior probability of one translation e' given f based on $\mathcal{H}(f)$.

4) $h_I(e, f)$: the length posterior feature of the specific target hypothesis with length I based on the mixture search space $\mathcal{H}(f)$ generated by the HM decoder:

$$h_I(e, f) = \sum_{e' \in \mathcal{H}(f), |e'| = |e| = I} P(e' | f)$$

Note here that features in $h_{\mathcal{H}(f)}(e, f)$ and $h_I(e, f)$ will be computed when the computations of all the remainder features in two categories have already finished for each e in $\mathcal{H}(f)$, and they will be used to update current HM decoding model scores.

Consensus features based on component search spaces have already shown effectiveness (Kumar et al., 2009; DeNero et al., 2010; Duan et al., 2010). We leverage consensus features based on the mixture search space newly generated in HM decoding as well. The length posterior feature (Zen and Ney, 2006) is used to adjust the preference of HM decoder for longer or shorter translations, and the stemmed *n*-gram posterior features are used to provide more discriminative power for HM decoding and to decrease the effects of morphological changes in words for more accurate computation of consensus statistics.

The second feature category contains a set of general features. Although there are more features that can be incorporated into HM decoding besides the ones we list below, we only utilize the most representative ones for convenience:

1) $h_{Length}(e, f)$: the word count feature.

2) $h_{LM}(e, f)$: the language model feature.

3) $h_{Dict}(e, f)$: the dictionary-based feature that counts how many lexicon pairs can be found in a given translation pair (e, f).

4) $h_{[\cdot]}(e, f)$ and $h_{<\cdot>}(e, f)$: reordering features that penalize the uses of straight and inverted BTG rules during the derivation of e in HM decoding. These two features are specific to BTG-based HM decoding (Section 2.4.1):

$$h_{[\cdot]}(e, f) = \sum_{r \in \mathcal{D}(e)} \delta_r([\cdot])$$

$$h_{<\cdot>}(e, f) = \sum_{r \in \mathcal{D}(e)} \delta_r(<\cdot>)$$

5) $h_{Rule}(e, f)$ and $h_{Glue}(e, f)$: reordering features that penalize the uses of hierarchical and glue rules during the derivation of e in HM decoding. These two features are specific to SCFG-based HM decoding (Section 2.4.2):

$$h_{Rule}(e, f) = \sum_{r \in \mathcal{D}(e)} \delta_r(R)$$

$$h_{Glue}(e, f) = \sum_{r \in \mathcal{D}(e)} \delta_r([\cdot])$$

R is the hierarchical rule set provided by the HM decoder itself, $\delta_r(R)$ equals to 1 when r is provided by R, and 0 otherwise.

6) $h_{New}(e, f)$: the feature that counts how many *n*-grams in e are newly generated by the HM decoder, which cannot be found in all existing component search spaces:

$$h_{New}(e, f) = \sum_{\omega \in e} \#_{\omega}(e) \bar{\delta}_{\omega}\left(\bigcup_{n=1}^{N} \mathcal{H}_n(f)\right)$$

$\bar{\delta}_{\omega}(\bigcup_{n=1}^{N} \mathcal{H}_n(f))$ equals to 1 when ω does not exist in $\bigcup_{n=1}^{N} \mathcal{H}_n(f)$, and 0 otherwise.

The MERT algorithm (Och, 2003) is used to tune weights of HM decoding features.

2.4 Decoding Algorithms

Two CKY-style algorithms for HM decoding are presented in this subsection. The first one is based on BTG (Wu, 1997), and the second one is based on SCFG, similar to Chiang (2005).

2.4.1 BTG-based HM Decoding

The first algorithm, *BTG-HMD*, is presented in Algorithm 1, where hypotheses of two consecutive source spans are composed using two BTG rules:

- *Straight rule* $[\cdot]$. It combines translations of two consecutive blocks into a single larger block in a straight order.
- *Inverted rule* $<\cdot>$. It combines translations of two consecutive blocks into a single larger block in an inverted order.

These two rules are used bottom-up until the whole source sentence is fully covered. We use two reordering rule penalty features, $h_{[\cdot]}(e,f)$ and $h_{<\cdot>}(e,f)$, to penalize the uses of these two rules.

Algorithm 1: BTG-based HM Decoding

```
1:  for each component model $\mathcal{M}_n$ do
2:      output the search space $\mathcal{H}_n(f)$ for the input $f$
3:  end for
4:  for $l = 1$ to $|f| - 1$ do
5:      for all $i, j$ s.t. $j - i = l$ do
6:          $\mathcal{H}(f_i^j) = \{\emptyset\}$
7:          for all $k$ s.t. $i \leq k < j$ do
8:              for $e_1 \in \mathcal{H}(f_i^k)$ and $e_2 \in \mathcal{H}(f_{k+1}^j)$ do
9:                  add $e = Comb_{[\cdot]}(e_1, e_2)$ to $\mathcal{H}(f_i^j)$
10:                 add $e = Comb_{<\cdot>}(e_1, e_2)$ to $\mathcal{H}(f_i^j)$
11:             end for
12:         end for
13:         for each hypothesis $e \in \bigcup_{n=1}^{N} \mathcal{H}_n(f_i^j)$ do
14:             compute HM decoding features for $e$
15:             add $e$ to $\mathcal{H}(f_i^j)$
16:         end for
17:         for each hypothesis $e \in \mathcal{H}(f_i^j)$ do
18:             compute the $n$-gram and length posterior
                features for $e$ based on $\mathcal{H}(f_i^j)$
19:             update current HM decoding score of $e$
20:         end for
21:     end for
22: end for
23: return $\hat{e} \in \mathcal{H}(f)$ with the maximum model score
```

In BTG-HMD, in order to derive translations for a source span f_i^j, we compose hypotheses of any two smaller spans f_i^k and f_{k+1}^j using two BTG rules in line 9 and 10, $Comb_r(e_1, e_2)$ denotes the operations that firstly combine e_1 and e_2 using one BTG rule r and secondly compute HM decoding features for the newly generated hypothesis e. We compute HM decoding features for hypotheses contained in all existing component search spaces $\bigcup_{n=1}^{N} \mathcal{H}_n(f_i^j)$ as well, and add them to $\mathcal{H}(f_i^j)$. From line 17 to 20, we update current HM decoding scores for all hypotheses in $\mathcal{H}(f_i^j)$ using the n-gram and length posterior features computed based on $\mathcal{H}(f_i^j)$. When the whole source sentence is fully covered, we return the hypothesis with the maximum model score as the final best translation.

2.4.2 SCFG-based HM Decoding

The second algorithm, *SCFG-HMD*, is presented in Algorithm 2. An additional rule set $\mathcal{R}$, which is provided by the HM decoder, is used to compose hypotheses. It includes hierarchical rules extracted using Chiang (2005)'s method and glue rules. Two reordering rule penalty features, $h_{Rule}(e,f)$ and $h_{Glue}(e,f)$, are used to adjust the preferences of using hierarchical rules and glue rules.

Algorithm 2: SCFG-based HM Decoding

```
1:  for each component model $\mathcal{M}_n$ do
2:      output the search space $\mathcal{H}_n(f)$ for the input $f$
3:  end for
4:  for $l = 1$ to $|f| - 1$ do
5:      for all $i, j$ s.t. $j - i = l$ do
6:          $\mathcal{H}(f_i^j) = \{\emptyset\}$
7:          for each rule $r \in \mathcal{R}$ that matches $f_i^j$ do
8:              for $e_1 \in \mathcal{H}(r_{\#_1})$ and $e_2 \in \mathcal{H}(r_{\#_2})$ do
9:                  add $e = Comb_r(e_1, e_2)$ to $\mathcal{H}(f_i^j)$
10:             end for
11:         end for
12:         for each hypothesis $e \in \bigcup_{n=1}^{N} \mathcal{H}_n(f_i^j)$ do
13:             compute HM decoding features for $e$
14:             add $e$ to $\mathcal{H}(f_i^j)$
15:         end for
16:         for each hypothesis $e \in \mathcal{H}(f_i^j)$ do
17:             compute the $n$-gram and length posterior
                features for $e$ based on $\mathcal{H}(f_i^j)$
18:             update current HM decoding score of $e$
19:         end for
20:     end for
21: end for
22: return $\hat{e} \in \mathcal{H}(f)$ with the maximum model score
```

Compared to BTG-HMD, the key differences in SCFG-HMD are located from line 7 to 11, where the translation for a given span f_i^j is generated by replacing the non-terminals in a hierarchical rule $r \in \mathcal{R}$ with their corresponding target translations, $r_{\#_i}$ is the source span that is covered by the i^{th} non-terminal of r, $\mathcal{H}(r_{\#_i})$ is the search space for $r_{\#_i}$ predicted by the HM decoder.

3 Comparisons to Related Techniques

3.1 Model Combination and Mixture Model based MBR Decoding

Model combination (DeNero et al., 2010) is an approach that selects translations from a conjoint search space using information from multiple SMT component models; Duan et al. (2010) presents a similar method, which utilizes a mixture model to combine distributions of hypotheses from different systems for Bayes-risk computation, and selects final translations from the combined search spaces using MBR decoding. Both of these two methods share a common limitation: they only re-rank the combined search space, without the capability to generate new translations. In contrast, by reusing hypotheses generated by all component systems in HM decoding, translations beyond any existing search space can be generated.

3.2 Co-Decoding and Joint Decoding

Li et al. (2009a) proposes collaborative decoding, an approach that combines translation systems by re-ranking partial and full translations iteratively using *n*-gram features from the predictions of other member systems. However, in co-decoding, all member systems must work in a synchronous way, and hypotheses between different systems cannot be shared during decoding procedure; Liu et al. (2009) proposes joint-decoding, in which multiple SMT models are combined in either translation or derivation levels. However, their method relies on the correspondence between nodes in hypergraph outputs of different models. HM decoding, on the other hand, can use hypotheses from component search spaces directly without any restriction.

3.3 Hybrid Decoding

Hybrid decoding (Cui et al., 2010) resembles our approach in the motivation. This method uses the system combination technique in decoding directly to combine partial hypotheses from different SMT models. However, confusion network construction brings high computational complexity. What's more, partial hypotheses generated by confusion network decoding cannot be assigned exact feature values for future use in higher level decoding, and they only use feature values of 1-best hypothesis as an approximation. HM decoding, on the other hand, leverages a set of enriched features, which are computable for all the hypotheses generated by either component systems or the HM decoder.

4 Experiments

4.1 Data and Metric

Experiments are conducted on the NIST Chinese-to-English MT tasks. The NIST 2004 (MT04) data set is used as the development set, and evaluation results are reported on the NIST 2005 (MT05), the newswire portions of the NIST 2006 (MT06) and 2008 (MT08) data sets. All bilingual corpora available for the NIST 2008 constrained data track of Chinese-to-English MT task are used as training data, which contain 5.1M sentence pairs, 128M Chinese words and 147M English words after pre-processing. Word alignments are performed using GIZA++ with the intersect-diag-grow refinement. The English side of bilingual corpus plus Xinhua portion of the LDC English Gigaword Version 3.0 are used to train a 5-gram language model.

Translation performance is measured in terms of case-insensitive BLEU scores (Papineni et al., 2002), which compute the brevity penalty using the shortest reference translation for each segment. Statistical significance is computed using the bootstrap re-sampling approach proposed by Koehn (2004). Table 1 gives some data statistics.

Data Set	#Sentence	#Word
MT04(dev)	1,788	48,215
MT05	1,082	29,263
MT06	616	17,316
MT08	691	17,424

Table 1: Statistics on dev and test data sets

4.2 Component Systems

For convenience of comparing HM decoding with several related decoding techniques, we include two state-of-the-art SMT systems as component systems only:

- *PB*. A phrase-based system (Xiong et al., 2006) with one lexicalized reordering model based on the maximum entropy principle.
- *DHPB*. A string-to-dependency tree-based system (Shen et al., 2008), which translates source strings to target dependency trees. A target dependency language model is used as an additional feature.

Phrasal rules are extracted on all bilingual data, hierarchical rules used in DHPB and reordering rules used in SCFG-HMD are extracted from a selected data set[3]. Reordering model used in PB is trained on the same selected data set as well. A trigram dependency language model used in DHPB is trained with the outputs from Berkeley parser on all language model training data.

4.3 Contrastive Techniques

We compare HM decoding with three multiple-system based decoding techniques:

- *Word-Level System Combination* (SC). We re-implement an IHMM alignment based system combination method proposed by Li et al. (2009b). The setting of the N-best candidates used is the same as the original paper.
- *Co-decoding* (CD). We re-implement it based on Li et al. (2009a), with the only difference that only two models are included in our re-implementation, instead of three in theirs. For each test set, co-decoding outputs three results, two for two member systems, and one for the further system combination.
- *Model Combination* (MC). Different from co-decoding, MC produces single one output for each input sentence. We re-implement this method based on DeNero et al. (2010) with two component models included.

4.4 Comparison to Component Systems

We compared HM decoding with two component SMT systems first (in Table 2). 30 features are used to annotate each hypothesis in HM decoding, including: 8 n-gram posterior features computed from PB/DHPB forests for $1 \leq n \leq 4$; 8 stemmed n-gram posterior features computed from stemmed PB/DHPB forests for $1 \leq n \leq 4$; 4 n-gram posterior features and 1 length posterior feature computed from the mixture search space of HM decoder for $1 \leq n \leq 4$; 1 LM feature; 1 word count feature; 1 dictionary-based feature; 2 grammar-specified rule penalty features for either BTG-HMD or SCFG-HMD; 4 count features for newly generated n-grams in HM decoding for $1 \leq n \leq 4$. All n-gram posteriors are computed using the efficient algorithm proposed by Kumar et al. (2009).

Model	BLEU%			
	MT04	MT05	MT06	MT08
PB	38.93	38.21	33.59	29.62
DHPB	39.90	39.76	35.00	30.43
BTG-HMD	**41.24***	**41.26***	**36.76***	**31.69***
SCFG-HMD	**41.31***	**41.19***	**36.63***	**31.52***

Table 2: HM decoding vs. single component system decoding (*: significantly better than each component system with $\boldsymbol{p} < 0.01$)

From table 2 we can see, both BTG-HMD and SCFG-HMD outperform decoding results of the best component system (DHPB) with significant improvements: +1.50, +1.76, and +1.26 BLEU points on MT05, MT06, and MT08 for BTG-HMD; +1.43, +1.63 and +1.09 BLEU points on MT05, MT06, and MT08 for SCFG-HMD. We also notice that BTG-HMD performs slight better than SCFG-HMD on test sets. We think the potential reason is that more reordering rules are used in SCFG-HMD to handle phrase movements than BTG-HMD do; however, current HM decoding model lacks the ability to distinguish the qualities of different rules.

We also investigate on the effects of different HM-decoding features. For the convenience of comparison, we divide them into five categories:

- *Set-1*. 8 n-gram posterior features based on 2 component search spaces plus 3 commonly used features (1 LM feature, 1 word count feature and 1 dictionary-based feature).
- *Set-2*. 8 stemmed n-gram posterior features based on 2 stemmed component search spaces.
- *Set-3*. 4 n-gram posterior features and 1 length posterior feature based on the mixture search space of the HM decoder.
- *Set-4*. 2 grammar-specified reordering rule penalty features.
- *Set-5*. 4 count features for unseen n-grams generated by HM decoder itself.

Except for the dictionary-based feature, all the features contained in Set-1 are used by the latest multiple-system based consensus decoding techniques (DeNero et al., 2010; Duan et al., 2010). We use them as the starting point. Each time, we add one more feature set and describe the changes of performances by drawing two curves for each HM decoding algorithm on MT08 in Figure 3.

[3] LDC2003E07, LDC2003E14, LDC2005T06, LDC2005T10, LDC2005E83, LDC2006E26, LDC2006E34, LDC2006E85 and LDC2006E92

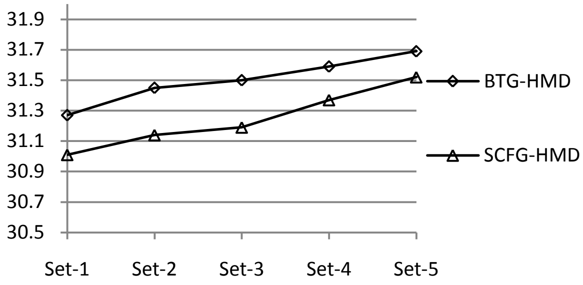

Figure 3: Effects of using different sets of HM decoding features on MT08

With Set-1 used only, HM-decoding has already outperformed the best component system, which shows the strong contributions of these features as proved in related work; small gains (+0.2 BLEU points) are achieved by using 8 stemmed *n*-gram posterior features in Set-2, which shows consensus statistics based on *n*-grams in their stem forms are also helpful; *n*-gram and length posterior features based on mixture search space bring improvements as well; reordering rule penalty features and count features for unseen *n*-grams boost newly generated hypotheses specific for HM decoding, and they contribute to the overall improvements.

4.5 Comparison to System Combination

Word-level system combination is state-of-the-art method to improve translation performance using outputs generated by multiple SMT systems. In this paper, we compare our HM decoding with the combination method proposed by Li et al. (2009b). Evaluation results are shown in Table 3.

Model	BLEU%			
	MT04	MT05	MT06	MT08
SC	41.14	40.70	36.04	31.16
BTG-HMD	**41.24**	**41.26**$^+$	**36.76**$^+$	**31.69**$^+$
SCFG-HMD	**41.31**$^+$	**41.19**$^+$	**36.63**$^+$	**31.52**$^+$

Table 3: HM decoding vs. system combination (+: significantly better than SC with $\boldsymbol{p} < 0.05$)

Compared to word-level system combination, both BTG-HMD and SCFG-HMD can provide significant improvements. We think the potential reason for these improvements is that, system combination can only use a small portion of the component systems' search spaces; HM decoding, on the other hand, can make full use of the entire translation spaces of all component systems.

4.6 Comparison to Consensus Decoding

Consensus decoding is another decoding technique that motivates our approach. We compare our HM decoding with two latest multiple-system based consensus decoding approaches, co-decoding and model combination. We list the comparison results in Table 4, in which CD-PB and CD-DHPB denote the translation results of two member systems in co-decoding respectively, CD-Comb denotes the results of further combination using outputs of CD-PB and CD-DHPB, MC denotes the results of model combination.

Model	BLEU%			
	MT04	MT05	MT06	MT08
CD-PB	40.39	40.34	35.20	30.39
CD-DHPB	40.81	40.56	35.73	30.87
CD-Comb	41.27	41.02	36.37	31.54
MC	41.19	40.96	36.30	31.43
BTG-HMD	**41.24**	**41.26**$^+$	**36.76**$^+$	**31.69**
SCFG-HMD	**41.31**	**41.19**	**36.63**$^+$	**31.52**

Table 4: HM decoding vs. consensus decoding (+: significantly better than the best result of consensus decoding methods with $\boldsymbol{p} < 0.05$)

Table 4 shows that after an additional system combination procedure, CD-Comb performs slight better than MC. Both BTG-HMD and SCFG-HMD perform consistent better than CD and MC on all blind test sets, due to its richer generative capability and usage of larger search spaces.

4.7 System Combination over BTG-HMD and SCFG-HMD Outputs

As BTG-HMD and SCFG-HMD are based on two different decoding grammars, we could perform system combination over the outputs of these two settings ($\mathrm{SC}^{\mathrm{BTG+SCFG}}$) for further improvements as well, just as Li et al. (2009a) did in co-decoding. We present evaluation results in Table 5.

Model	BLEU%			
	MT04	MT05	MT06	MT08
BTG-HMD	41.24	41.26	36.76	31.69
SCFG-HMD	41.31	41.19	36.63	31.52
$\mathrm{SC}^{\mathrm{BTG+SCFG}}$	**41.74**$^+$	**41.53**$^+$	**37.11**$^+$	**32.06**$^+$

Table 5: System combination based on the outputs of BTG-HMD and SCFG-HMD (+: significantly better than the best HM decoding algorithm (SCFG-HMD) with $\boldsymbol{p} < 0.05$)

After system combination, translation results are significantly better than all decoding approaches investigated in this paper: up to 2.11 BLEU points over the best component system (DHPB), up to 1.07 BLEU points over system combination, up to 0.74 BLEU points over co-decoding, and up to 0.81 BLEU points over model combination.

4.8 Evaluation of Oracle Translations

In the last part, we evaluate the quality of oracle translations on the *n*-best lists generated by HM decoding and all decoding approaches discussed in this paper. Oracle performances are obtained using the metric of sentence-level BLEU score proposed by Ye et al. (2007), and each decoding approach outputs its 1000-best hypotheses, which are used to extract oracle translations.

Model	BLEU%			
	MT04	MT05	MT06	MT08
PB	49.53	48.36	43.69	39.39
DHPB	50.66	49.59	44.68	40.47
SC	51.77	50.84	46.87	42.11
CD-PB	50.26	50.10	45.65	40.52
CD-DHPB	51.91	50.61	46.23	41.01
CD-Comb	52.10	51.00	46.95	42.20
MC	52.03	51.22	46.60	42.23
BTG-HMD	**52.69**$^+$	**51.75**$^+$	**47.08**	**42.71**$^+$
SCFG-HMD	**52.94**$^+$	**51.40**	**47.27**$^+$	**42.45**$^+$
SC$^{BTG+SCFG}$	**53.58**$^+$	**52.03**$^+$	**47.90**$^+$	**43.07**$^+$

Table 6: Oracle performances of different methods (+: significantly better than the best multiple-system based decoding method (CD-Comb) with $p < 0.05$)

Results are shown in Table 6: compared to each single component system, decoding methods based on multiple SMT systems can provide significant improvements on oracle translations; word-level system combination, collaborative decoding and model combination show similar performances, in which CD-Comb performs best; BTG-HMD, SCFG-HMD and SC$^{BTG+SCFG}$ can obtain significant improvements than all the other approaches, and SC$^{BTG+SCFG}$ performs best on all evaluation sets.

5 Conclusion

In this paper, we have presented the hypothesis mixture decoding approach to combine multiple SMT models, in which hypotheses generated by multiple component systems are used to compose new translations. HM decoding method integrates the advantages of both system combination and consensus decoding techniques into a unified framework. Experimental results across different NIST Chinese-to-English MT evaluation data sets have validated the effectiveness of our approach.

In the future, we will include more SMT models and explore more features, such as syntax-based features, helping to improve the performance of HM decoding. We also plan to investigate more complicated reordering models in HM decoding.

References

David Chiang. 2005. A Hierarchical Phrase-based Model for Statistical Machine Translation. In *Proceedings of the Association for Computational Linguistics*, pages 263-270.

David Chiang. 2010. Learning to Translate with Source and Target Syntax. In *Proceedings of the Association for Computational Linguistics*, pages 1443-1452.

Lei Cui, Dongdong Zhang, Mu Li, Ming Zhou, and Tiejun Zhao. 2010. Hybrid Decoding: Decoding with Partial Hypotheses Combination over Multiple SMT Systems. In *Proceedings of the International Conference on Computational Linguistics*, pages 214-222.

John DeNero, David Chiang, and Kevin Knight. 2009. Fast Consensus Decoding over Translation Forests. In *Proceedings of the Association for Computational Linguistics*, pages 567-575.

John DeNero, Shankar Kumar, Ciprian Chelba and Franz Och. 2010. Model Combination for Machine Translation. In *Proceedings of the North American Association for Computational Linguistics*, pages 975-983.

Nan Duan, Mu Li, Dongdong Zhang, and Ming Zhou. 2010. Mixture Model-based Minimum Bayes Risk Decoding using Multiple Machine Translation Systems. In *Proceedings of the International Conference on Computational Linguistics*, pages 313-321.

Michel Galley, 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 *Proceedings of the Association for Computational Linguistics*, pages 961-968.

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

Philipp Koehn. 2004. Statistical Significance Tests for Machine Translation Evaluation. In *Proceedings of the Conference on Empirical Methods on Natural Language Processing*, pages 388-395.

Shankar Kumar and William Byrne. 2004. Minimum Bayes-Risk Decoding for Statistical Machine Translation. In *Proceedings of the North American Association for Computational Linguistics*, pages 169-176.

Shankar Kumar, Wolfgang Macherey, Chris Dyer, and Franz Och. 2009. Efficient Minimum Error Rate Training and Minimum Bayes-Risk Decoding for Translation Hypergraphs and Lattices. In *Proceedings of the Association for Computational Linguistics*, pages 163-171.

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

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

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

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

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

Kishore Papineni, Salim Roukos, Todd Ward, and Weijing Zhu. 2002. BLEU: a method for automatic evaluation of machine translation. In *Proceedings of the Association for Computational Linguistics*, pages 311-318.

Libin Shen, Jinxi Xu, and Ralph Weischedel. 2008. A new String-to-Dependency Machine Translation Algorithm with a Target Dependency Language Model. In *Proceedings of the Association for Computational Linguistics*, pages 577-585.

Antti-Veikko Rosti, Spyros Matsoukas, and Richard Schwartz. 2007. Improved Word-Level System Combination for Machine Translation. In *Proceedings of the Association for Computational Linguistics*, pages 312-319.

Roy Tromble, Shankar Kumar, Franz Och, and Wolfgang Macherey. 2008. Lattice Minimum Bayes-Risk Decoding for Statistical Machine Translation. In *Proceedings of the Conference on Empirical Methods on Natural Language Processing*, pages 620-629.

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

Deyi Xiong, Qun Liu, and Shouxun Lin. 2006. Maximum Entropy based Phrase Reordering Model for Statistical Machine Translation. In *Proceedings of the Association for Computational Linguistics*, 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 *Proceedings of the Second Workshop on Statistical Machine Translation*, pages 240-247.