Hypergraph Training and Decoding of System Combination in SMT

Yupeng Liu Tiejun Zhao Sheng Li
Harbin Institute of Technology
ypliu@mtlab.hit.edu.cn
tjzhao@mtlab.hit.edu.cn
lisheng@mtlab.hit.edu.cn

Abstract

Tranditional n-best based training and decoding method of system combination can propogate the error because of imprecision parameter estimation and too early prunning. In order to alleviate the problem, the paper proposes hypergraph (HG) based three-pass training and three-pass decoding for different features. In order to construct HG, this paper introduces simplified bracket transduction grammar (SBTG) into HG based system combination. At three-pass decoding, decoder uses each-pass features to generate target translation. At three-pass training, we introduce minimum risk (MR) with deterministic annealing (DA) into the training criterion in order to overcome overfitting, and furthermore compare two training procedures: minimum error training (MERT) on *n*-best and MR&DA on HG. The unified training and decoding approaches of HG based system combination outperform baseline using conventional Cube Prunning on Chinese-to-English benchmark corpus NIST08 test set.

1 Introduction

System combination has been proven that consensus translations are usually better than the translations of individual systems (Matusov et. al., 2006; Rosti et. al., 2007). Confusion network (CN) (Matusov et al. 2006 and Rosti et al. 2007) for word-level combination is a widely adopted approach for combining SMT output, which can significantly outperform sentence-level re-ranking methods and phrase-level combination (Rosti et. al., 2007). During constructing CN, word alignment between skeleton/backbone and hypothesis and skeleton selection are two key issues in this approach. To solve first issue, Translation Error Rate (TER) (Snover et al., 2006) based alignment was proposed in Sim et al. (2007); IHMM (He et al., 2008) got the better alignment using source language as pivot language; ITG-based alignment (Karakos et al., 2008) uses the ITG constrain during constructing CN; lattice-based system combination (Feng et al., 2009) normalized the alignment between the skeleton and the hypothesis into the lattice without breaking the phrase structure; incremental strategy (Rosti et al., 2008; Li et al., 2009) was added into the monolingual alignment algorithm including TER and IHMM in order to avoid pairwise alignment error. To solve second problem, joint optimization (He and Toutanova, 2009) integrated CN construction and decoding into a decoder without skeleton selection; multiple CNs was first proposed in (Matusov et al.2006; Rosti et al., 2007), and was implemented via combining several different hypothesis alignment metrics (Du and Way, 2009).

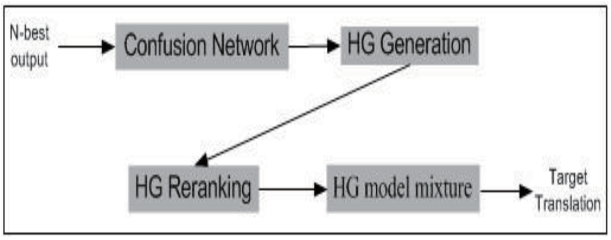

Figure 1: The pipeline of three-pass training and decoding for HG generation, HG reranking and HG model mixture

However, there are few works about training and decoding for system combination. Tranditional n-best training method train feature weights at limited hypothesis space and propogate the errors to target translation. These errors will severely hurt the translation quanlity. To alleviate such error, a HG was applied to many areas, such as translation rule extraction (Mi and Huang, 2008; Tu et al., 2010), model training (Li and Eisner, 2009b), decoding (Liu et al., 2009; Li et al., 2009a; Kumar et al., 2009; DeNero et al., 2009) in the field of machine translation, constituent parsing (Huang and Chiang, 2005). Overall, HG gives large search space for training and decoding which is expected to avoid the search error caused by the imprecision parameter estimation and early prunning.

This paper introduces HG into system combination and explores to address three problems:

(1) Since the HG technology gives better performance than conventional training and decoding method in many natural language processing areas, would the technology still be efficient in system combination?

(2) Models in SMT exhibit spurious ambiguous (Li et al. 2009).We can resolve it by using the re-estimation of *n*-gram probability on HG. Does HG based system combination model have the same problem as models in SMT?

(3) During constructing CN, different system combination models use the different construction strategy. Are these models of system combination complementary?

To answer these questions, we will mix models of system combination through three-pass decoding and three-pass training (Figure 1) for different features, which are firstly for HG generation, secondly for HG re-ranking according to three types of *n*-gram probability and finally for mixing the HG model of system combination. MR with/without DA on HG or not, which attempts to solve the increasingly difficult optimization problem, is introduced into training procedure.

This paper is structured as follows. After introducing the definition of SBTG on HG of system combination in section 2, we first, in the section 3, show training criterion on HG, and then in section 4, we give the details about several n-gram probability computation methods on HG decoding. In section 5, experiment results and analysis are presented. In section 6, we review the related work of this paper.

2 SBTG on HG

Formally, a HG in system combination is defined as a 4-tuple $H=<V, E, G, \mathbf{R}>$, where V is a finite set of hypernode, E is a finite set of hyperedge, $G \in \mathbf{R}$ is the unique goal item in H, and $\mathbf{R}$ is a set of weights. For a input sentence of target language $e_1^J = e_1, \dots e_J$, each hypernode is in the form of X_i^j, which denotes the partial translation of target partial language $e_i, \dots e_j$ spanning the substring from $i-1$ to j. Each hyperedge $e \in E$ is a triple tuple $e=<T(e), h(e), w(e)>$, where $T(e) \in V$ is a vector of tail nodes, $h(e) \in V$ is its head, and $w(e)$ is a weight function from $\mathbf{R}^{|T(e)|}$ to $\mathbf{R}$.

Our HG-based system combination is represented by simplified bracket transduction grammar (SBTG). Formally, the set of these hyperedges can be defined as a 3-tuple $E=<T, N, P>$, where T is a set of the terminal word symbol in target language, N is a set of the non-terminal symbol including three symbols $N=\{S, X_1, X_2\}$, P is a set of production rules including two types:

- Lexical rule: $X \rightarrow w,\ w \in D$
- Non-terminal rule : $S \rightarrow X_1\ X_2$,
  $X \rightarrow X_1\ X_2$

where D is a dictionary including *null* word (ε) for normalization in system combination, start symbol ($G \in R$) and single word. Non-terminal rule is like straight ordering in bracket transduction grammar (BTG).

For example, suppose we have one skeleton “he gives me an apple” and one hypothesis “he gave me apple”. We can obtain the tabular form (Li et al. 2009) of CN in Figure 2(a) through some alignment metrics. We can obtain CN in the lower part of Figure 2(b) using some alignment metric. Figure 2(b) show the generation of HG through a set of production rule (lexical and nonterminal rule) in SBTG. The hyperedges between nodes denote the decision steps that produce head node from tail nodes. For example, the incoming hyperedge of the hypernode *<”he … gave”, 0-2>* could correspond to two lexical rules and a nonterminal rule in SBTG; the two incoming hyperedges of the hypernode *<”he … apple”, 0-5>* could correspond to two non-terminal rules in SBTG; the one incoming hyperedge of the hypernode *<”S”, 0-5>*could

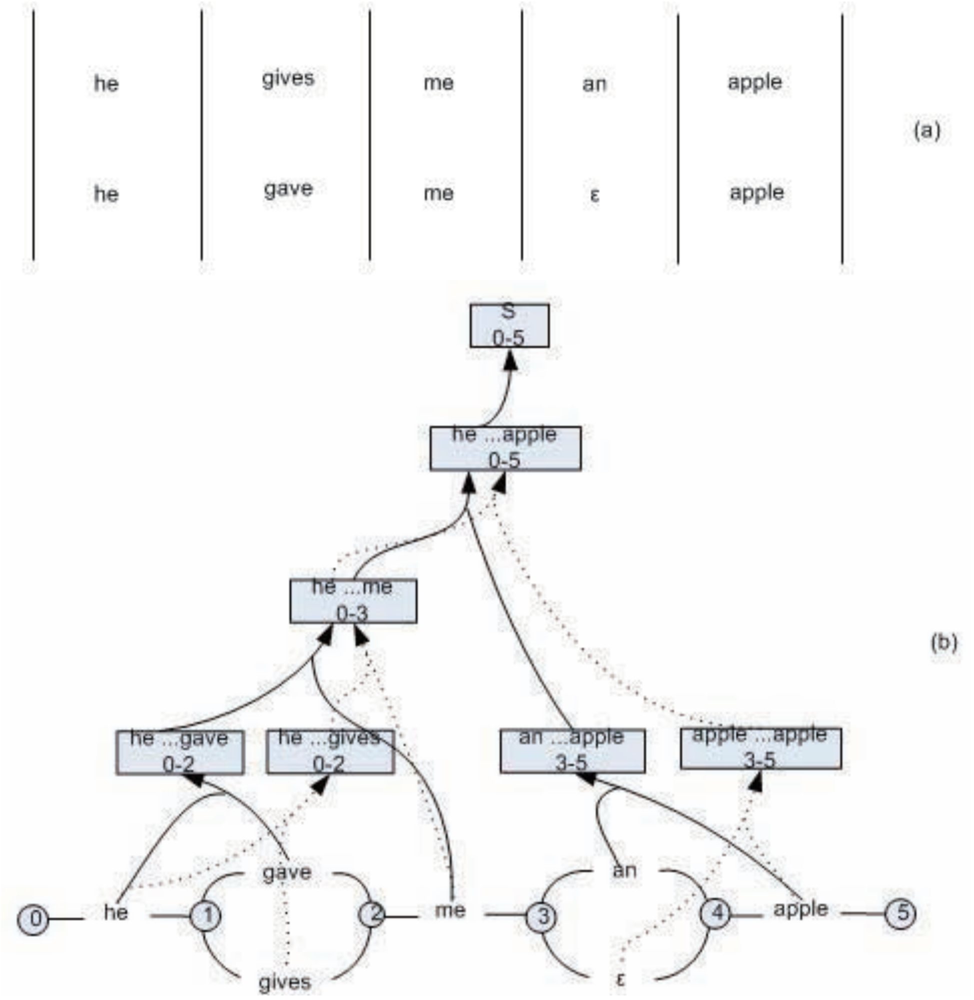

Figure 2: (a) the tabular form of confusion network (b) a HG produced by rules in SBTG. Solid and dashed lines denote the alternative rule for the same hypernode. If we use 2-gram language model, 1-gram left and right equivalent state is introduced into the hypernode.

correspond to *S*→*X1 X2* in SBTG. Using *S*→*X1 X2* in SBTG is for decoding convenience. *S* and *X* are two non-terminal in SBTG.

3 Decoding on HG

To integrate HG into system combination, we model the derivation generation using the probabilistic SBTG. The viterbi decoding for latent variable model can be formalized as:

$$\begin{aligned}\hat{e} &= \arg\max_{e} \max_{d\in D} p_{\lambda,\theta}(e,d \mid f) \\ &= \arg\max_{e} \max_{d\in D} \frac{\exp(\lambda \sum_i \theta_i h_i(e,f,d))}{\sum_{e,d} \exp(\lambda \sum_i \theta_i h_i(e,f,d))} \quad (1) \\ &= \arg\max_{e} \max_{d\in D} \exp(\lambda \sum_i \theta_i h_i(e,f,d))\end{aligned}$$

where D is the set of all derivations generated by a set of production rules, d is one such derivation, $\sum_{e,d} exp\ (\lambda \sum_i \theta_i h_i(e,f,d))$ is a nomalization constant, $h_i(e,f,d)$is the i-th feature, θ_i is the feature weight of i-th feature, $\hat{e}$ is the best translation through searching the HG. Hyperparameter λ is shown in section 4. Generic features in decoder are the same as Rosti et al. (2007), inculding word posterior of every system, language model, ε penalty and word penalty. At first-pass decoding, we find the best target translation using generic features. At second-pass and third-pass decoding, the best translation can be obtained using n-gram probability (in section 3.1) and mixture factor (in section 3.2) features respectively.

Algorithm 1 Computation of N-gram Model/Posterior/Count Expectation

```
run inside and outside algorithm
compute hyperedge posterior probability $p(e|H)$
for $v \in H$   ▷ for each topological hypernode
  initialize the quantities $I_v(w_n)$, $c(w_n)$ and $c(h(w_n))$ of hypernode $v$
    for $e \in IN(v)$ ▷ for each incoming  hyperedge
    initialize the quantities $\bar{b}(w_n)$ and $b$ of hyperedge $e$
        $b = p(e) \times I_{ant(v)}$
        $\bar{b}(w_n) = p(e) \times (I_{ant(v)} - I_{ant(v)}(w_n))$
        if $w_n \in e$
            $c(w_n)$ += $p(e|H)$
            $c(h(w_n))$ += $p(e|H)$
            $I_v(w_n)$ += $b$
        else
            $I_v(w_n) += b - \bar{b}(w_n)$
$p(w_n) = \frac{c(w_n)}{c(h(w_n))}$
$q(w_n) = \frac{I_{root}(w_n)}{I_{root}}$
return n-gram model score $p(w_n)$
       n-gram count expectation $c(w_n)$
       n-gram posterior probability $q(w_n)$
```

Figure 3: the Computation of N-gram Model/Count Expectation/Posterior

3.1 N-GRAM Probability

We present an extension of the algorithm in Li et al. (2009a) that allows us to efficiently compute n-gram probability encoded in HGs. We employ three types of *n*-gram estimation probability in Figure 3, and then compare these probabilities. The algorithm can compute them including *n*-gram model (Li et al.2009a) that is just as unsmothed n-gram probability in language model, n-gram count expectation (DeNero et al., 2009) that is the expectation of n-gram count, n-gram posterior (Kumar et al. 2009; DeNero et al., 2010) that is the ratio of n-gram inside score and regular inside score of root.

This algorithm1 can in principle compute n-gram model, count expectation and posterior probability on HG. For each hypernode, we track four quantities:

(1) the regular inside scores I_v that sum the scores of all derivations rooted at *v* and can be computed by inside recursion procedure;

(2) *n*-gram inside scores $I_v(w_n)$ that sum the scores of all derivations rooted at *v* that contain *n*-gram w_n;

(3) soft count $c(w_n)$ and $c(h(w_n))$ that sum the posterior probabilities of all hyperedges introducing w_n or $h(w_n)$ into HG.

For each hyperedge, we track two quantities: *b* that sum the scores for all derivations through the hyperedge *e*; $\bar{b}(w_n)$ that sum the scores for derivations that don't contain w_n through the hyperedge *e*. We have two probability form about the hyperedge: one for the posterior probability $p(e|H)$ (Li et al. 2009) of the hyperedge, the other for the weight $p(e)$ of the hyperedge. *ant*(*v*) and $h(w_n)$ denote the antecedent of hypernode *v* and (*n-1*)-gram prefix of *n*-gram word w_n respectively.

In order to construct hypernode, we merge hyperedge of the same left and right equivalent *(n-1)*-gram state (*n* is the order of language model we use) into the same hypernode, which already removeε .

3.2 Mixture Factor

Rosti et al., 2007 first proposed using multiple CN even though Matusov et al., 2006 proposed it first. Multiple CN consists of many individual CNs which are constructed based on the same alignment metric. Super CN (Du and Way, 2009) has the same idea with multiple CNs, but it uses the different alignment metric for each CN.

The idea of multiple CNs proves the complementarities of system combination models. We follow the complementary from another way that combine identical *n*-gram and model probability from each HG of system combination.The decoding formulation of mixture model is as follow:

$$\hat{d} = \arg\max_{d} \left[\sum_{i=1}^{I} \left(\sum_{n=1}^{N} (w_i^n h_i^n(e,f,d)) + w_i^v v_i(e,f,d) \right) + w^l l(e,f,d) \right] \quad (2)$$

We use a linear model with three types of feature functions of a derivation: *n*-gram probability $h_i^n(e,f,d)$ scores a derivation of each component model according to the n-grams it contains; viterbi model score $v_i^n(e,f,d)$ is the dot product of the vectors of generic feature value and weight; length function $l(e,f,d)$ is the length of a derivation of each model in system combination. *I* is the number of system, and *N* is *n*-gram number. If we have two models of system combination and use the *5*-gram probability model, we have 2*5+2+1=13 mixture factor numbers. The optimization of mixture factor is at third-pass training and third-pass decoding. We initially construct the hypergraph bottom-up. After the construction, we use lazy Algorithm 3 (Huang and Chiang, 2005) to generate k-best translations in three-pass decoding.

4 MR Training on HG

To overcome the overfitting, Smith and Eisner (2006) smoothed the risk function by DA to ensure the search space is as large as possible before objective function achieves the optimal weight. The objective function can be defined as follow:

$$L = \sum_{e,d \in HG} R_H(p_{\lambda,\theta}) - T \cdot E_H(p_{\lambda,\theta}) \quad (3)$$

where $p_{\lambda,\theta}$ is definded in equation (1), $E_H(p_{\lambda,\theta})$ is the entropy of probability distribution $p_{\lambda,\theta}$, and λ is the scaling hyperparameter, when $\lambda = 0$ giving the uniform distribution; when $\lambda = 1$ giving the original model probability distribution; and as $\lambda \to \infty$, the probability approaches the winner-take-all Viterbi function, and θ is feature weight vector.

In order to optimize by gradient descending, we need the gradient of two types of parameter. We use the first- and sencond-order semiring(Li et al. 2009b) to compute them. Both of them can be defined as follow:

$$\nabla L = \sum_{e,d \in HG} \nabla R_H(p_{\lambda,\theta}) - T \cdot \nabla E_H(p_{\lambda,\theta}) \quad (4)$$

where $T>0$ is a temperature parameter which is gradually lowered as the optimization progresses according to some annealing schedule. We perform the optimization in two steps: first optimizing θ_k; second optimizing λ. So the optimizer could exactly compensate for the increase of λ by decreasing the θ vector proportionately.

5 Experiments

We use NIST MT06 data set including 1099 sentences as the development set and NIST MT08 data set including 1357 sentences from both newswire and web-data genres as the test set. To save computation effort, the result on the development and test set are reported in case-insensitive BLEU score. The above system generates the 10-best of every sentence as input of system combination through the max-BLEU training (MERT). The language model used for the model is a *5*-gram model trained with Xinhua portion of LDC English Gigaword corpus version 3. The parameter and distortion model of incremental IHMM were set as Li et al. (2009). The lexical translation probabilities used in semantic similarity model are from a small portion (FBIS+GALE) of the constrained track training data. The skeleton is selected by minimum bayesian risk (MBR) and the loss function is BLEU.

Alignment	NIST06	NIST08
Worst Single System	27.33	21.45
Best Single System	32.60	27.75
Inc TER	38.21	31.35
Inc IHMM	39.34	32.82

Table 1: The result of single and system combination on the development and test set

We combine outputs of eight SMT systems[1]. Table 1 shows the performance of single system and combination system. Compared to the worst and best single system, incremental TER yield a large improvement (+5.61~+10.88 BLEU) on development set and (+3.6~+9.9 BLEU) on test set. Incremental IHMM achieve better performance (up to +1.13 and 1.47 BLEU score on the development and test set) than incremental TER.

We compare each-pass decoding of system combination based on HG. We report performance using incremental IHMM (Li et al., 2009) during first two-pass decoding. The components of mixture model in last-pass decoding are incremental TER and incremental IHMM.

[1] The input of system combination is the same as Li et al. (2009).

5.1 First-pass HG Decoding

During first-pass decoding, HG decoding with stack size 500 outperform the baseline (incremental IHMM) by +0.28 and +0.57 BLEU point on development and test set. Incremental IHMM model use Cube Prunning (Chiang 2007) and HG decoding use Cube Growing (Huang and Chiang 2007). Mixing model mixes the output of incremental IHMM and HG decoding model.

Model	NIST06	NIST08
Inc IHMM	39.34	32.82
HG Decoding	39.47	33.02
Mixing	39.62	33.39

Table 2: The result of first-pass HG decoding on the development and test set

5.2 Second-Pass HG Decoding

During second-pass decoding, we use the same beam size as first pass decoding because the outside probability estimation of the second-pass decoding is discriminative enough to guide second-pass HG Decoding. We develop a unified algorithm of three n-gram probabilty which are n-gram model (denoted by ngram_1), n-gram count expection (denoted by ngram_2) and n-gram posterior probability (denoted by ngram_3), and then compare the performance of them.

The Effect of *n*-gram Model: as shown in Table 3, decoding with *1-5*-gram+*wp* (word penalty denoted by *wp*) model of different estimation methods improve (+0.66, +0.35 and +0.29 BLEU score) over first-pass training and decoding (beam size is 500) on the development set, and we achieve an absolute improvement (+0.59, +0.60 and +0.56 BLEU score) on the test set. The experimental result proves that *n*-gram feature is effective.

The effect of Viterbi model can be seen through comparing Table 4 with Table 3. The various interpolation models show an improvement of +0.25, +0.3 and +0.27 BLEU points over model without Viterbi on the development set, and +0.25, +0.13 and +0.04 BLEU point on test set. By comparing with one-pass HG decoding, the best performance of three types of *n*-gram probability can improve by +0.91 and +0.84 BLEU score on the development and test set respectively. If we compare it with baseline (incremental IHMM), the best performance of three types of *n*-gram probability can be obtained when the setting is *Vi*+*1*-*5*gram_1+*wp*. It obtains +1.19 and +1.41 BLEU score on the development and test set respectively.

The experimental results prove the effeciency of *n*-gram and Viterbi+*n*-gram model.

n-gram model	NIST06	NIST08
1-*5*gram_1+*wp*	40.28	33.98
1-*5*gram_2+*wp*	39.97	33.99
1-*5*gram_3+*wp*	39.91	33.95

Table 3: The quality of second-pass decoding on the development and test set

Viterbi+*n*-gram	NIST06	NIST08
Vi+*1*-*5*gram_1+*wp*	40.53	34.23
Vi+*1*-*5*gram_2+*wp*	40.27	34.12
Vi+*1*-*5*gram_3+*wp*	40.18	33.99

Table 4: The quality of second-pass decoding with Viterbi baseline on the development and test set

The Effect of MR with DA: We compare the five training schema: MERT vs. MR with different setting, which are with/without DA, with/without quenching scaling factor λ and on HG. With the entropy constrains, starting temperature *T=1000*; quenching temperature *T=0.001*. The temperature is cooled by half at each step; then we double λ at each step. Once *T* is quite cool, it is common in practice to switch to rising λ directly and rapidly until some convergence condition. We optimize feature weight vector $\boldsymbol{\theta}$ and hyperparameter λ through BFGS optimization.

The configures of the experiment use the interpolation between *1*-*5*gram_1 and Viterbi model. We compare five settings on the development in Figure 4 and the test set in Table 5. MERT, MR without DA&quenching, MR&DA without quenching, MR&DA with quenching and MR&DA with quenching on HG achieve a BLEU score of 40.53, 40.17, 40.37, 40.50 and 40.50 on the development set. The best performance can be obtained by MERT on the development set, and meanwhile the worst performance can be obtained by it on test set. The fact proves the overfitting of MERT. The reason of decreased performance of 2-th iteration MR with DA is the intialization bias.

Training Criterion	NIST08
MERT	34.23

MR without DA&queching	34.24
MR&DA without quenching	34.28
MR&DA with quenching	34.29
MR&DA with quenching on HG	34.29

Table 5: The MERT and MR with/without DA performance of the test set

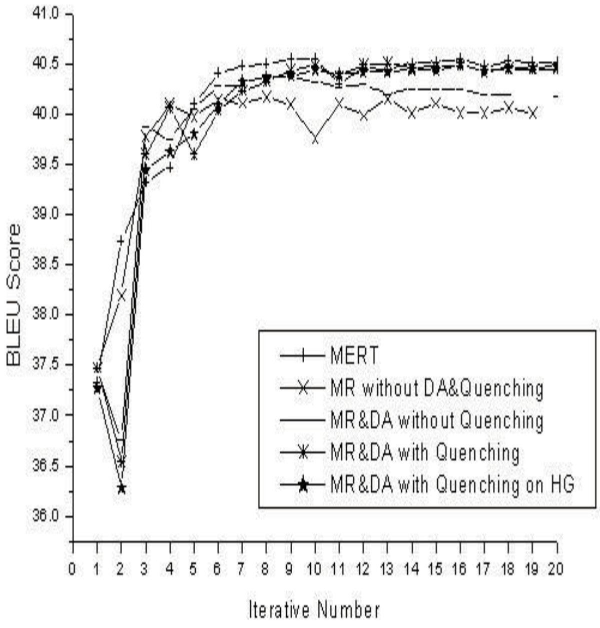

Figure 4: The MERT and MR with/without DA performance on the development set

Compared to MERT, MR&DA on HG has almost the same performance on test set because of a small number of features (Li and Eisner, 2009b) or a sparse feature of the non-terminal rule which only includes language model probability. In total, MR&DA on HG outperform baseline (incremental IHMM) using Cube Prunning up to +1.47 in BLEU score.

5.3 Third-Pass HG Decoding

Firstly, the n-gram features are extracted from incremental TER and IHMM model via second-pass decoding. Then, we mix both n-grams on one of HG. Finally, we can search the HG during third-pass decoding.

Model	NIST06	NIST08
Incremental TER	39.99	33.19
Incremental IHMM	40.50	34.29
Mixture Model (MR&DA)	40.37	34.41
Mixture Model (MR&DA on HG)	40.39	34.42

Table 6: The performance comparison of MR&DA with/without HG on the development and test set

Compared to incremental IHMM and TER model after two-pass decoding, mixture model of MR&DA on HG in Table 6 achieves +0.13 and +0.23 BLEU point improvement. Though the third-pass doesn’t yield better improvement, we conclude that there is little complementary between two models or the consensus are already modeled by CN construction. In total, the three-pass training and decoding outperform baseline up to +1.6 BLEU point.

6 Related Work

A unified framework (Pauls et al., 2009, Arun et al., 2010) was employed in MBR training and decoding. However, their methods aren’t based on the HG. In this paper, we present a unified framework of training and decoding on HG. On the other hand, there are several research on HG based decoding (Li et al., 2009a; Kumar et al., 2009; Denero et al., 2009), which use the *n*-gram probability to further improve the performance of the single system. In this paper, we compare three n-gram probability.

In the view of HG mixture, our method is most similar to the mixture model based on HG in SMT. Duan et al. (2010) proposed a two-pass parameter optimization: first for *n*-gram probability weight for each system; second for mixture model weight whose number is the same as the number of system. DeNero et al. (2010) employed one-pass training for tuning the weight of *n*-gram posterior and model score, and don’t achieve the best tuning effect of each search space on model score. There are two major differences between our approach and above two approaches. Firstly, our model has three-pass training phase. Secondly, we have many weights for every *n*-gram probability and Vitebi score in each involved component model.

7 Conclusion and Future Work

In this paper, we present a system combination based on HG. Comparing to conventional training and decoding method, our method on HG uses more features to refine the expressive ability of the model. We have empirically verified the success on HG of system combination in three aspects: HG decoding, HG re-ranking and mixture model.

References

Abhishek Arun, Barry Haddow, and Philipp Koehn. 2010. A Unified Approach to Minimum Risk Training and Decoding. In Proceedings of 5th Workshop on Statistical Machine Translation.

David Chiang. 2007. Hierarchical Phrase-based Translation. Computational Linguistics, 33(2).

John Denero, David Chiang, and Kevin Knight. 2009. Fast Consensus Decoding over Translation Forest. In proceedings of ACL.

John DeNero, Shankar Kumar, Ciprian Chelba, and Franz Och. 2010. Model Combination for Machine Translation. In proceedings of NAACL.

Nan Duan, Li Mu, Dongdong Zhang, and Ming Zhou. 2010. Mixture Model-based Minimum Bayes Risk Decoding using Multiple Machine Translation Systems. In Proceedings of Coling.

Jinhua Du and Andy Way. 2009. A Three-pass System Combination Framework by Combining Multiple Hypothesis Alignment Methods. In Proceedings of the International Conference on Asian Language Processing (IALP 2009), Singapore,December, pages 172-176.

Yang Feng, Yang Liu, Haitong Mi,Qun Liu, and Yajuan Lu. 2009. Lattice-based System Combination for Statistical Machine Translation. In Proceedings of ACL.

Damianos Karakos, Jason Eisner, Sanjeev Khudanpur, and Markus Dreyer. 2008. Machine Translation System Combination using ITG-based Alignments. In Proceedings of ACL.

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 ACL. pages. 163-171.

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 Proc. of EMNLP.

Xiaodong He and Kristina Toutanova. 2009. Joint Optimization for Machine Translation System Combination. In Proc. of EMNLP.

Liang Huang and David Chiang. 2005. Better k-best Parsing. In Proceedings of the International Workshop on Parsing Technologies (IWPT), pages 53–64.

Liang Huang and David Chiang. 2007. Forest Rescoring: Faster Decoding with Integrated Language Models. In Proceedings of ACL, Prague, Czech Rep.

Liang Huang. 2008. Forest reranking: Discriminative parsing with Non-local Features. In Proc. of ACL/HLT.

Chi-Ho Li, Xiaodong He,Yupeng Liu, and Ning Xi. 2009. Incremental HMM Alignment for MT System Combination. In Proceedings of ACL.

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. of ACL, pages 585-592.

Zhifei Li, Jason Eisner and Sanjeev Khudanpur. 2009a. Variational Decoding for Statistical Machine Translation. In Proceedings of ACL.

Zhifei Li and Jason Eisner. 2009b. First- and Second-order Expectation Semirings with Applications to Minimum-Risk Training on Translation Forests. In Proceedings of EMNLP.

Yang Liu, Haitao Mi, Yang Feng, and Qun Liu. 2009 Joint Decoding with Multiple Translation Models. In Proc. of ACL, pages 576-584.

Evgeny Matusov, Nicola Ueffing, and Hermann Ney. 2006. Computing Consensus Translation from Multiple Machine Translation Systems using Enhanced Hypothesis Alignment. In Proceedings of EACL.

Haitao Mi and Liang Huang. 2008. Forest-based Translation Rule Extraction. In Proc. Of EMNLP.

Adam Pauls, John DeNero, and Dan Klein. 2009. Consensus Training for Consensus Decoding in Machine Translation. In Proceedings of EMNLP.

Antti-Veikko I. Rosti, Spyros Matsoukas, and Richard Schwartz. 2007. Improved Word-level System Combination for Machine Translation. In Proceedings of ACL.

Antti-Veikko I. Rosti, Bing Zhang, Spyros Matsoukas, and Richard Schwartz. 2008. Incremental Hypothesis Alignment for Building Confusion Networks with Application to Machine Translation System Combination.In Proceedings of the 3rd ACL Workshop on SMT.

Matthew Snover, Bonnie Dorr, Rich Schwartz, Linnea Micciulla, and John Makhoul 2006. A Study of Translation Edit Rate with Targeted Human Annotation. In Proceedings of AMTA.

Khe Chai Sim, William J. Byrne, Mark J.F. Gales, Hichem Sahbi, and Phil C. Woodland. 2007. Consensus Network Decoding for Statistical Machine Translation System Combination. In Proc. of ICASSP, pages 105–108.

David A. Smith and Jason Eisner. 2006. Minimum Risk Annealing for Training Log-linear Models. In Proceedings of COLING-ACL, pages 787–794.

Zhaopeng Tu, Yang Liu, Young-Sook Hwang, Qun Liu, and Shouxun Lin. 2010. Dependency Forest for Statistical Machine Translation. In Proceedings of COLING.