Indirect-HMM-based Hypothesis Alignment for Combining Outputs from Machine Translation Systems

Xiaodong He[†], Mei Yang[‡] [*], Jianfeng Gao[†], Patrick Nguyen[†], and Robert Moore[†]

[†]Microsoft Research
One Microsoft Way
Redmond, WA 98052 USA
`{xiaohe,jfgao,panguyen,`
`bobmoore}@microsoft.com`

[‡]Dept. of Electrical Engineering
University of Washington
Seattle, WA 98195, USA
`yangmei@u.washington.edu`

Abstract

This paper presents a new hypothesis alignment method for combining outputs of multiple machine translation (MT) systems. An indirect hidden Markov model (IHMM) is proposed to address the synonym matching and word ordering issues in hypothesis alignment. Unlike traditional HMMs whose parameters are trained via maximum likelihood estimation (MLE), the parameters of the IHMM are estimated *indirectly* from a variety of sources including word semantic similarity, word surface similarity, and a distance-based distortion penalty. The IHMM-based method significantly outperforms the state-of-the-art TER-based alignment model in our experiments on NIST benchmark datasets. Our combined SMT system using the proposed method achieved the best Chinese-to-English translation result in the constrained training track of the 2008 NIST Open MT Evaluation.

1 Introduction

System combination has been applied successfully to various machine translation tasks. Recently, confusion-network-based system combination algorithms have been developed to combine outputs of multiple machine translation (MT) systems to form a consensus output (Bangalore, et al. 2001, Matusov et al., 2006, Rosti et al., 2007, Sim et al., 2007). A confusion network comprises a sequence of sets of alternative words, possibly including *null*'s, with associated scores. The consensus output is then derived by selecting one word from each set of alternatives, to produce the sequence with the best overall score, which could be assigned in various ways such as by voting, by

using posterior probability estimates, or by using a combination of these measures and other features.

Constructing a confusion network requires choosing one of the hypotheses as the backbone (also called "skeleton" in the literature), and other hypotheses are aligned to it at the word level. High quality hypothesis alignment is crucial to the performance of the resulting system combination. However, there are two challenging issues that make MT hypothesis alignment difficult. First, different hypotheses may use different synonymous words to express the same meaning, and these synonyms need to be aligned to each other. Second, correct translations may have different word orderings in different hypotheses and these words need to be properly reordered in hypothesis alignment.

In this paper, we propose an indirect hidden Markov model (IHMM) for MT hypothesis alignment. The HMM provides a way to model both synonym matching and word ordering. Unlike traditional HMMs whose parameters are trained via maximum likelihood estimation (MLE), the parameters of the IHMM are estimated *indirectly* from a variety of sources including word semantic similarity, word surface similarity, and a distance-based distortion penalty, without using large amount of training data. Our combined SMT system using the proposed method gave the best result on the Chinese-to-English test in the constrained training track of the 2008 NIST Open MT Evaluation (MT08).

2 Confusion-network-based MT system combination

The current state-of-the-art is confusion-network-based MT system combination as described by

---

Rosti and colleagues (Rosti et al., 2007a, Rosti et al., 2007b). The major steps are illustrated in Figure 1. In Fig. 1 (a), hypotheses from different MT systems are first collected. Then in Fig. 1 (b), one of the hypotheses is selected as the backbone for hypothesis alignment. This is usually done by a sentence-level minimum Bayes risk (MBR) method which selects a hypothesis that has the minimum average distance compared to all hypotheses. The backbone determines the word order of the combined output. Then as illustrated in Fig. 1 (c), all other hypotheses are aligned to the backbone. Note that in Fig. 1 (c) the symbol ε denotes a *null* word, which is inserted by the alignment normalization algorithm described in section 3.4. Fig. 1 (c) also illustrates the handling of synonym alignment (e.g., aligning "car" to "sedan"), and word re-ordering of the hypothesis. Then in Fig. 1 (d), a confusion network is constructed based on the aligned hypotheses, which consists of a sequence of sets in which each word is aligned to a list of alternative words (including *null*) in the same set. Then, a set of global and local features are used to decode the confusion network.

E_1 he have good car	$E_B = \arg\min_{E' \in \mathbf{E}} \sum_{E \in \mathbf{E}} TER(E', E)$
E_2 he has nice sedan	
E_3 it a nice car	e.g., $E_B = E_1$
E_4 a sedan he has	
(a) hypothesis set	(b) backbone selection

E_B he have ε good car	he have ε good car	
(crossing lines)	he has ε nice sedan	
	it ε a nice car	
E_4 a ε sedan he has	he has a ε sedan	
(c) hypothesis alignment	(d) confusion network	
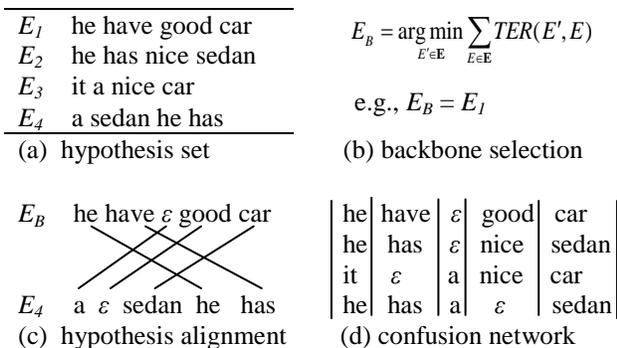
Figure 1: Confusion-network-based MT system combination.

3 Indirect-HMM-based Hypothesis Alignment

In confusion-network-based system combination for SMT, a major difficulty is aligning hypotheses to the backbone. One possible statistical model for word alignment is the HMM, which has been widely used for bilingual word alignment (Vogel et al., 1996, Och and Ney, 2003). In this paper, we propose an indirect-HMM method for monolingual hypothesis alignment.

3.1 IHMM for hypothesis alignment

Let $e_1^I = (e_1, ..., e_I)$ denote the backbone, $e_1'^J = (e_1', ..., e_J')$ a hypothesis to be aligned to e_1^I, and $a_1^J = (a_1, ..., a_J)$ the alignment that specifies the position of the backbone word aligned to each hypothesis word. We treat each word in the backbone as an HMM state and the words in the hypothesis as the observation sequence. We use a first-order HMM, assuming that the emission probability $p(e_j' | e_{a_j})$ depends only on the backbone word, and the transition probability $p(a_j | a_{j-1}, I)$ depends only on the position of the last state and the length of the backbone. Treating the alignment as hidden variable, the conditional probability that the hypothesis is generated by the backbone is given by

$$p(e_1'^J | e_1^I) = \sum_{a_1^J} \prod_{j=1}^{J} \left[p(a_j | a_{j-1}, I) p(e_j' | e_{a_j}) \right] \quad (1)$$

As in HMM-based bilingual word alignment (Och and Ney, 2003), we also associate a *null* with each backbone word to allow generating hypothesis words that do not align to any backbone word.

In HMM-based hypothesis alignment, emission probabilities model the similarity between a backbone word and a hypothesis word, and will be referred to as the similarity model. The transition probabilities model word reordering, and will be called the distortion model.

3.2 Estimation of the similarity model

The similarity model, which specifies the emission probabilities of the HMM, models the similarity between a backbone word and a hypothesis word. Since both words are in the same language, the similarity model can be derived based on both semantic similarity and surface similarity, and the overall similarity model is a linear interpolation of the two:

$$p(e_j' | e_i) = \alpha \cdot p_{sem}(e_j' | e_i) + (1 - \alpha) \cdot p_{sur}(e_j' | e_i) \quad (2)$$

where $p_{sem}(e'_j | e_i)$ and $p_{sur}(e'_j | e_i)$ reflect the semantic and surface similarity between e'_j and e_i, respectively, and α is the interpolation factor.

Since the semantic similarity between two target words is source-dependent, the semantic similarity model is derived by using the source word sequence as a hidden layer:

$$
\begin{aligned}
&p_{sem}(e'_j | e_i) \\
&= \sum_{k=0}^{K} p(f_k | e_i) p(e'_j | f_k, e_i) \\
&\approx \sum_{k=0}^{K} p(f_k | e_i) p(e'_j | f_k)
\end{aligned}
\tag{3}
$$

where $f_1^K = (f_1, ..., f_K)$ is the source sentence. Moreover, in order to handle the case that two target words are synonyms but neither of them has counter-part in the source sentence, a *null* is introduced on the source side, which is represented by f_0. The last step in (3) assumes that first e_i generates all source words including *null*. Then e_j' is generated by all source words including *null*.

In the common SMT scenario where a large amount of bilingual parallel data is available, we can estimate the translation probabilities from a source word to a target word and vice versa via conventional bilingual word alignment. Then both $p(f_k | e_i)$ and $p(e'_j | f_k)$ in (3) can be derived:

$$
p(e'_j | f_k) = p_{s2t}(e'_j | f_k)
$$

where $p_{s2t}(e'_j | f_k)$ is the translation model from the source-to-target word alignment model, and $p(f_k | e_i)$, which enforces the sum-to-1 constraint over all words in the source sentence, takes the following form,

$$
p(f_k | e_i) = \frac{p_{t2s}(f_k | e_i)}{\sum_{k=0}^{K} p_{t2s}(f_k | e_i)}
$$

where $p_{t2s}(f_k | e_i)$ is the translation model from the target-to-source word alignment model. In our method, $p_{t2s}(null | e_i)$ for all target words is

simply a constant p_{null}, whose value is optimized on held-out data [1].

The surface similarity model can be estimated in several ways. A very simple model could be based on exact match: the surface similarity model, $p_{sur}(e'_j | e_i)$, would take the value 1.0 if $e' = e$, and 0 otherwise [2]. However, a smoothed surface similarity model is used in our method. If the target language uses alphabetic orthography, as English does, we treat words as letter sequences and the similarity measure can be the length of the longest matched prefix (LMP) or the length of the longest common subsequence (LCS) between them. Then, this raw similarity measure is transformed to a surface similarity score between 0 and 1 through an exponential mapping,

$$
p_{sur}(e'_j | e_i) = \exp\left\{ \rho \cdot \left[s(e'_j, e_i) - 1 \right] \right\}
\tag{4}
$$

where $s(e'_j, e_i)$ is computed as

$$
s(e'_j, e_i) = \frac{M(e'_j, e_i)}{\max(|e'_j|, |e_i|)}
$$

and $M(e'_j, e_i)$ is the raw similarity measure of e_j' e_i, which is the length of the LMP or LCS of e_j' and e_i. and ρ is a smoothing factor that characterizes the mapping, Thus as ρ approaches infinity, $p_{sur}(e'_j | e_i)$ backs off to the exact match model. We found the smoothed similarity model of (4) yields slightly better results than the exact match model. Both LMP- and LCS- based methods achieve similar performance but the computation of LMP is faster. Therefore, we only report results of the LMP-based smoothed similarity model.

3.3 Estimation of the distortion model

The distortion model, which specifies the transition probabilities of the HMM, models the first-order dependencies of word ordering. In bilingual HMM-based word alignment, it is commonly assumed that transition probabilities

---

[1] The other direction, $p_{s2t}(e'_i | null)$, is available from the source-to-target translation model.
[2] Usually a small back-off value is assigned instead of 0.

$p(a_j = i \mid a_{j-1} = i', I)$ depend only on the jump distance $(i - i')$ (Vogel et al., 1996):

$$p(i \mid i', I) = \frac{c(i - i')}{\sum\limits_{l=1}^{I} c(l - i')} \qquad (5)$$

As suggested by Liang et al. (2006), we can group the distortion parameters $\{c(d)\}$, $d = i - i'$, into a few buckets. In our implementation, 11 buckets are used for $c(\leq-4)$, $c(-3)$, ... $c(0)$, ..., $c(5)$, $c(\geq6)$. The probability mass for transitions with jump distance larger than 6 and less than -4 is uniformly divided. By doing this, only a handful of $c(d)$ parameters need to be estimated. Although it is possible to estimate them using the EM algorithm on a small development set, we found that a particularly simple model, described below, works surprisingly well in our experiments.

Since both the backbone and the hypothesis are in the same language, It seems intuitive that the distortion model should favor monotonic alignment and only allow non-monotonic alignment with a certain penalty. This leads us to use a distortion model of the following form, where K is a tuning factor optimized on held-out data.

$$c(d) = \left(1 + |d - 1|\right)^{-\kappa}, \; d = -4, \ldots, 6 \qquad (6)$$

As shown in Fig. 2, the value of distortion score peaks at $d=1$, i.e., the monotonic alignment, and decays for non-monotonic alignments depending on how far it diverges from the monotonic alignment.
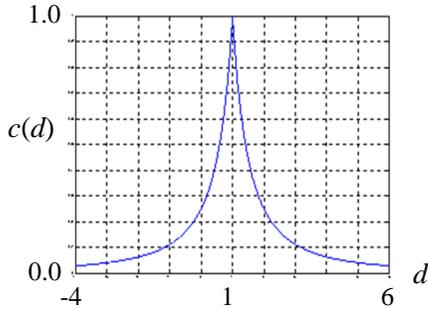

Figure 2, the distance-based distortion parameters computed according to (6), where K=2.

Following Och and Ney (2003), we use a fixed value p_0 for the probability of jumping to a *null* state, which can be optimized on held-out data, and the overall distortion model becomes

$$\tilde{p}(i \mid i', I) = \begin{cases} p_0 & \text{if } i = \textit{null} \text{ state} \\ (1 - p_0) \cdot p(i \mid i', I) & \text{otherwise} \end{cases}$$

3.4 Alignment normalization

Given an HMM, the Viterbi alignment algorithm can be applied to find the best alignment between the backbone and the hypothesis,

$$\hat{a}_1^J = \arg\max_{a_1^J} \prod_{j=1}^{J} \left[p(a_j \mid a_{j-1}, I) p(e_j' \mid e_{a_j}) \right] \qquad (7)$$

However, the alignment produced by the algorithm cannot be used directly to build a confusion network. There are two reasons for this. First, the alignment produced may contain 1-N mappings between the backbone and the hypothesis whereas 1-1 mappings are required in order to build a confusion network. Second, if hypothesis words are aligned to a *null* in the backbone or vice versa, we need to insert actual *null*s into the right places in the hypothesis and the backbone, respectively. Therefore, we need to normalize the alignment produced by Viterbi search.
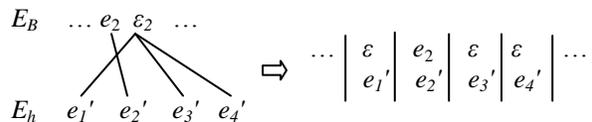
(a) hypothesis words are aligned to the backbone *null*
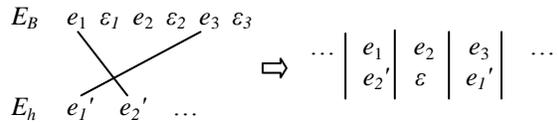
(b) a backbone word is aligned to no hypothesis word

Figure 3: illustration of alignment normalization

First, whenever more than one hypothesis words are aligned to one backbone word, we keep the link which gives the highest occupation probability computed via the forward-backward algorithm. The other hypothesis words originally

aligned to the backbone word will be aligned to the *null* associated with that backbone word.

Second, for the hypothesis words that are aligned to a particular *null* on the backbone side, a set of *null*s are inserted around that backbone word associated with the *null* such that no links cross each other. As illustrated in Fig. 3 (a), if a hypothesis word e_2' is aligned to the backbone word e_2, a *null* is inserted in front of the backbone word e_2 linked to the hypothesis word e_1' that comes before e_2'. *Null*s are also inserted for other hypothesis words such as e_3' and e_4' after the backbone word e_2. If there is no hypothesis word aligned to that backbone word, all *null*s are inserted after that backbone word .[3]

For a backbone word that is aligned to no hypothesis word, a null is inserted on the hypothesis side, right after the hypothesis word which is aligned to the immediately preceding backbone word. An example is shown in Fig. 3 (b).

4 Related work

The two main hypothesis alignment methods for system combination in the previous literature are GIZA++ and TER-based methods. Matusov et al. (2006) proposed using GIZA++ to align words between different MT hypotheses, where all hypotheses of the test corpus are collected to create hypothesis pairs for GIZA++ training. This approach uses the conventional HMM model bootstrapped from IBM Model-1 as implemented in GIZA++, and heuristically combines results from aligning in both directions. System combination based on this approach gives an improvement over the best single system. However, the number of hypothesis pairs for training is limited by the size of the test corpus. Also, MT hypotheses from the same source sentence are correlated with each other and these hypothesis pairs are not i.i.d. data samples. Therefore, GIZA++ training on such a data set may be unreliable.

Bangalore et al. (2001) used a multiple string-matching algorithm based on Levenshtein edit distance, and later Sim et al. (2007) and Rosti et al. (2007) extended it to a TER-based method for hypothesis alignment. TER (Snover et al., 2006)

measures the minimum number of edits, including substitution, insertion, deletion, and shift of blocks of words, that are needed to modify a hypothesis so that it exactly matches the other hypothesis. The best alignment is the one that gives the minimum number of translation edits. TER-based confusion network construction and system combination has demonstrated superior performance on various large-scale MT tasks (Rosti. et al, 2007). However, when searching for the optimal alignment, the TER-based method uses a strict surface hard match for counting edits. Therefore, it is not able to handle synonym matching well. Moreover, although TER-based alignment allows phrase shifts to accommodate the non-monotonic word ordering, all non-monotonic shifts are penalized equally no matter how short or how long the move is, and this penalty is set to be the same as that for substitution, deletion, and insertion edits. Therefore, its modeling of non-monotonic word ordering is very coarse-grained.

In contrast to the GIZA++-based method, our IHMM-based method has a similarity model estimated using bilingual word alignment HMMs that are trained on a large amount of bi-text data. Moreover, the surface similarity information is explicitly incorporated in our model, while it is only used implicitly via parameter initialization for IBM Model-1 training by Matusov et al. (2006). On the other hand, the TER-based alignment model is similar to a coarse-grained, non-normalized version of our IHMM, in which the similarity model assigns no penalty to an exact surface match and a fixed penalty to all substitutions, insertions, and deletions, and the distortion model simply assigns no penalty to a monotonic jump, and a fixed penalty to all other jumps, equal to the non-exact-match penalty in the similarity model.

There have been other hypothesis alignment methods. Karakos, et al. (2008) proposed an ITG-based method for hypothesis alignment, Rosti et al. (2008) proposed an incremental alignment method, and a heuristic-based matching algorithm was proposed by Jayaraman and Lavie (2005).

5 Evaluation

In this section, we evaluate our IHMM-based hypothesis alignment method on the Chinese-to-English (C2E) test in the constrained training track

---

[3] This only happens if no hypothesis word is aligned to a backbone word but some hypothesis words are aligned to the *null* associated with that backbone word.

of the 2008 NIST Open MT Evaluation (NIST, 2008). We compare to the TER-based method used by Rosti et al. (2007). In the following experiments, the NIST BLEU score is used as the evaluation metric (Papineni et al., 2002), which is reported as a percentage in the following sections.

5.1 Implementation details

In our implementation, the backbone is selected with MBR. Only the top hypothesis from each single system is considered as a backbone. A uniform posteriori probability is assigned to all hypotheses. TER is used as loss function in the MBR computation.

Similar to (Rosti et al., 2007), each word in the confusion network is associated with a word posterior probability. Given a system S, each of its hypotheses is assigned with a rank-based score of $1/(1+r)^{\eta}$, where r is the rank of the hypothesis, and η is a rank smoothing parameter. The system specific rank-based score of a word w for a given system S is the sum of all the rank-based scores of the hypotheses in system S that contain the word w at the given position (after hypothesis alignment). This score is then normalized by the sum of the scores of all the alternative words at the same position and from the same system S to generate the system specific word posterior. Then, the total word posterior of w over all systems is a sum of these system specific posteriors weighted by system weights.

Beside the word posteriors, we use language model scores and a word count as features for confusion network decoding.

Therefore, for an M-way system combination that uses N LMs, a total of $M+N+1$ decoding parameters, including M-1 system weights, one rank smoothing factor, N language model weights, and one weight for the word count feature, are optimized using Powell's method (Brent, 1973) to maximize BLEU score on a development set[4].

Two language models are used in our experiments. One is a trigram model estimated from the English side of the parallel training data, and the other is a 5-gram model trained on the English GigaWord corpus from LDC using the MSRLM toolkit (Nguyen et al, 2007).

---

[4] The parameters of IHMM are not tuned by maximum-BLEU training.

In order to reduce the fluctuation of BLEU scores caused by the inconsistent translation output length, an unsupervised length adaptation method has been devised. We compute an expected length ratio between the MT output and the source sentences on the development set after maximum-BLEU training. Then during test, we adapt the length of the translation output by adjusting the weight of the word count feature such that the expected output/source length ratio is met. In our experiments, we apply length adaptation to the system combination output at the level of the whole test corpus.

5.2 Development and test data

The development (dev) set used for system combination parameter training contains 1002 sentences sampled from the previous NIST MT Chinese-to-English test sets: 35% from MT04, 55% from MT05, and 10% from MT06-newswire. The test set is the MT08 Chinese-to-English "current" test set, which includes 1357 sentences from both newswire and web-data genres. Both dev and test sets have four references per sentence.

As inputs to the system combination, 10-best hypotheses for each source sentence in the dev and test sets are collected from each of the eight single systems. All outputs on the MT08 test set were true-cased before scoring using a log-linear conditional Markov model proposed by Toutanova et al. (2008). However, to save computation effort, the results on the dev set are reported in case insensitive BLEU (ciBLEU) score instead.

5.3 Experimental results

In our main experiments, outputs from a total of eight single MT systems were combined. As listed in Table 1, Sys-1 is a tree-to-string system proposed by Quirk et al., (2005); Sys-2 is a phrase-based system with fast pruning proposed by Moore and Quirk (2008); Sys-3 is a phrase-based system with syntactic source reordering proposed by Wang et al. (2007a); Sys-4 is a syntax-based pre-ordering system proposed by Li et. al. (2007); Sys-5 is a hierarchical system proposed by Chiang (2007); Sys-6 is a lexicalized re-ordering system proposed by Xiong et al. (2006); Sys-7 is a two-pass phrase-based system with adapted LM proposed by Foster and Kuhn (2007); and Sys-8 is

a hierarchical system with two-pass rescoring using a parser-based LM proposed by Wang et al., (2007b). All systems were trained within the confines of the constrained training condition of NIST MT08 evaluation. These single systems are optimized with maximum-BLEU training on different subsets of the previous NIST MT test data. The bilingual translation models used to compute the semantic similarity are from the word-dependent HMMs proposed by He (2007), which are trained on two million parallel sentence-pairs selected from the training corpus allowed by the constrained training condition of MT08.

5.3.1 Comparison with TER alignment

In the IHMM-based method, the smoothing factor for surface similarity model is set to $\rho = 3$, the interpolation factor of the overall similarity model is set to $\alpha = 0.3$, and the controlling factor of the distance-based distortion parameters is set to $K=2$. These settings are optimized on the dev set. Individual system results and system combination results using both IHMM and TER alignment, on both the dev and test sets, are presented in Table 1. The TER-based hypothesis alignment tool used in our experiments is the publicly available TER Java program, TERCOM (Snover et al., 2006). Default settings of TERCOM are used in the following experiments.

On the dev set, the case insensitive BLEU score of the IHMM-based 8-way system combination output is about 5.8 points higher than that of the best single system. Compared to the TER-based method, the IHMM-based method is about 1.5 BLEU points better. On the MT08 test set, the IHMM-based system combination gave a case sensitive BLEU score of 30.89%. It outperformed the best single system by 4.7 BLEU points and the TER-based system combination by 1.0 BLEU points. Note that the best single system on the dev set and the test set are different. The different single systems are optimized on different tuning sets, so this discrepancy between dev set and test set results is presumably due to differing degrees of mismatch between the dev and test sets and the various tuning sets.

Table 1. Results of single and combined systems on the dev set and the MT08 test set

System	Dev ciBLEU%	MT08 BLEU%
System 1	34.08	21.75
System 2	33.78	20.42
System 3	34.75	21.69
System 4	37.85	25.52
System 5	37.80	24.57
System 6	37.28	24.40
System 7	32.37	25.51
System 8	34.98	26.24
TER	42.11	29.89
IHMM	43.62	30.89

In order to evaluate how well our method performs when we combine more systems, we collected MT outputs on MT08 from seven additional single systems as summarized in Table 2. These systems belong to two groups. Sys-9 to Sys-12 are in the first group. They are syntax-augmented hierarchical systems similar to those described by Shen et al. (2008) using different Chinese word segmentation and language models. The second group has Sys-13 to Sys-15. Sys-13 is a phrasal system proposed by Koehn et al. (2003), Sys-14 is a hierarchical system proposed by Chiang (2007), and Sys-15 is a syntax-based system proposed by Galley et al. (2006). All seven systems were trained within the confines of the constrained training condition of NIST MT08 evaluation.

We collected 10-best MT outputs only on the MT08 test set from these seven extra systems. No MT outputs on our dev set are available from them at present. Therefore, we directly adopt system combination parameters trained for the previous 8-way system combination, except the system weights, which are re-set by the following heuristics: First, the total system weight mass 1.0 is evenly divided among the three groups of single systems: {Sys-1~8}, {Sys-9~12}, and {Sys-13~15}. Each group receives a total system weight mass of 1/3. Then the weight mass is further divided in each group: in the first group, the original weights of systems 1~8 are multiplied by 1/3; in the second and third groups, the weight mass is evenly distributed within the group, i.e., 1/12 for each system in group 2, and 1/9 for each

system in group 3[5]. Length adaptation is applied to control the final output length, where the same expected length ratio of the previous 8-way system combination is adopted.

The results of the 15-way system combination are presented in Table 3. It shows that the IHMM-based method is still about 1 BLEU point better than the TER-based method. Moreover, combining 15 single systems gives an output that has a NIST BLEU score of 34.82%, which is 3.9 points better than the best submission to the NIST MT08 constrained training track (NIST, 2008). To our knowledge, this is the best result reported on this task.

Table 2. Results of seven additional single systems on the NIST MT08 test set

System	MT08 BLEU%
System 9	29.59
System 10	29.57
System 11	29.64
System 12	29.85
System 13	25.53
System 14	26.04
System 15	29.70

Table 3. Results of the 15-way system combination on the NIST MT08 C2E test set

Sys. Comb.	MT08 BLEU%
TER	33.81
IHMM	34.82

5.3.2 Effect of the similarity model

In this section, we evaluate the effect of the semantic similarity model and the surface similarity model by varying the interpolation weight α of (2). The results on both the dev and test sets are reported in Table 4. In one extreme case, $\alpha = 1$, the overall similarity model is based only on semantic similarity. This gives a case insensitive BLEU score of 41.70% and a case sensitive BLEU score of 28.92% on the dev and test set, respectively. The accuracy is significantly improved to 43.62% on the dev set and 30.89% on test set when $\alpha = 0.3$. In another extreme case, $\alpha =$

0, in which only the surface similarity model is used for the overall similarity model, the performance degrades by about 0.2 point. Therefore, the surface similarity information seems more important for monolingual hypothesis alignment, but both sub-models are useful.

Table 4. Effect of the similarity model

	Dev ciBLEU%	Test BLEU%
$\alpha = 1.0$	41.70	28.92
$\alpha = 0.7$	42.86	30.50
$\alpha = 0.5$	43.11	30.94
$\alpha = 0.3$	43.62	30.89
$\alpha = 0.0$	43.35	30.73

5.3.3 Effect of the distortion model

We investigate the effect of the distance-based distortion model by varying the controlling factor K in (6). For example, setting $K=1.0$ gives a linear-decay distortion model, and setting $K=2.0$ gives a quadratic smoothed distance-based distortion model. As shown in Table 5, the optimal result can be achieved using a properly smoothed distance-based distortion model.

Table 5. Effect of the distortion model

	Dev ciBLEU%	Test BLEU%
$K=1.0$	42.94	30.44
$K=2.0$	43.62	30.89
$K=4.0$	43.17	30.30
$K=8.0$	43.09	30.01

6 Conclusion

Synonym matching and word ordering are two central issues for hypothesis alignment in confusion-network-based MT system combination. In this paper, an IHMM-based method is proposed for hypothesis alignment. It uses a similarity model for synonym matching and a distortion model for word ordering. In contrast to previous methods, the similarity model explicitly incorporates both semantic and surface word similarity, which is critical to monolingual word alignment, and a smoothed distance-based distortion model is used to model the first-order dependency of word ordering, which is shown to be better than simpler approaches.

---

[5] This is just a rough guess because no dev set is available. We believe a better set of system weights could be obtained if MT outputs on a common dev set were available.

Our experimental results show that the IHMM-based hypothesis alignment method gave superior results on the NIST MT08 C2E test set compared to the TER-based method. Moreover, we show that our system combination method can scale up to combining more systems and produce a better output that has a case sensitive BLEU score of 34.82, which is 3.9 BLEU points better than the best official submission of MT08.

Acknowledgement

References

Srinivas Bangalore, German Bordel, and Giuseppe Riccardi. 2001. Computing consensus translation from multiple machine translation systems. In *Proc. of IEEE ASRU*, pp. 351–354.

Richard Brent, 1973. Algorithms for Minimization without Derivatives. Prentice-Hall, Chapter 7.

David Chiang. 2007. Hierarchical phrase-based translation. *Computational Linguistics*, 33(2):201–228.

George Foster and Roland Kuhn. 2007. Mixture-Model Adaptation for SMT. In *Proc. of the Second ACL Workshop on Statistical Machine Translation*. pp. 128 – 136.

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 *Proc. of COLING-ACL*, pp. 961–968.

Xiaodong He. 2007. Using Word-Dependent Transition Models in HMM based Word Alignment for Statistical Machine Translation. In *Proc. of the Second ACL Workshop on Statistical Machine Translation*.

Shyamsundar Jayaraman and Alon Lavie. 2005. Multi-engine machine translation guided by explicit word matching. In *Proc. of EAMT*. pp. 143 – 152.

Damianos Karakos, Jason Eisner, Sanjeev Khudanpur, and Markus Dreyer. 2008. Machine Translation System Combination using ITG-based Alignments. In *Proc. of ACL-HLT*, pp. 81–84.

Chi-Ho Li, Minghui Li, Dongdong Zhang, Mu Li, Ming Zhou, Yi Guan. 2007. A Probabilistic Approach to Syntax-based Reordering for Statistical Machine Translation. In *Proc. of ACL*. pp. 720 – 727.

Percy Liang, Ben Taskar, and Dan Klein. 2006. Alignment by Agreement. In *Proc. of NAACL*. pp 104 – 111.

Evgeny Matusov, Nicola Ueffing, and Hermann Ney. 2006. Computing consensus translation from multiple machine translation systems using enhanced hypotheses alignment. In *Proc. of EACL*, pp. 33–40.

Robert Moore and Chris Quirk. 2007. Faster Beam-Search Decoding for Phrasal Statistical Machine Translation. In *Proc. of MT Summit XI*.

Patrick Nguyen, Jianfeng Gao and Milind Mahajan. 2007. MSRLM: a scalable language modeling toolkit. *Microsoft Research Technical Report MSR-TR-2007-144.*

NIST. 2008. The 2008 NIST Open Machine Translation Evaluation. www.nist.gov/speech/tests/mt/2008/doc/

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

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. BLEU: a method for automatic evaluation of machine translation. In *Proc. of ACL*, pp. 311–318.

Koehn, Philipp, Franz Josef Och, and Daniel Marcu. 2003. Statistical phrase based translation. In *Proc. of NAACL*. pp. 48 – 54.

Chris Quirk, Arul Menezes, and Colin Cherry. 2005. Dependency treelet translation: Syntactically informed phrasal SMT. In *Proc. of ACL*. pp. 271–279.

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

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

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 *Proc. of the Third ACL Workshop on Statistical Machine Translation*, pp. 183–186.

Libin Shen, Jinxi Xu, Ralph Weischedel. 2008. A New String-to-Dependency Machine Translation Algorithm with a Target Dependency Language Model. In *Proc. of ACL-HLT*, pp. 577–585.

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*, *vol. 4*. pp. 105–108.

Matthew Snover, Bonnie Dorr, Rich Schwartz, Linnea Micciulla, and John Makhoul. 2006. A study of translation edit rate with targeted human annotation. In *Proc. of AMTA*.

Kristina Toutanova, Hisami Suzuki and Achim Ruopp. 2008. Applying Morphology Generation Models to Machine Translation. In *Proc. of ACL*. pp. 514 – 522.

Stephan Vogel, Hermann Ney, and Christoph Tillmann. 1996. HMM-based Word Alignment In Statistical Translation. In *Proc. of COLING*. pp. 836-841.

Chao Wang, Michael Collins, and Philipp Koehn. 2007a. Chinese Syntactic Reordering for Statistical Machine Translation. In *Proc. of EMNLP-CoNLL*. pp. 737-745.

Wen Wang, Andreas Stolcke, Jing Zheng. 2007b. Reranking Machine Translation Hypotheses With Structured and Web-based Language Models. In *Proc. of IEEE ASRU*. pp. 159 – 164.

Deyi Xiong, Qun Liu and Shouxun Lin. 2006. Maximum Entropy Based Phrase Reordering Model for Statistical Machine Translation. In *Proc. of ACL*. pp. 521 – 528.