Lattice-based System Combination for Statistical Machine Translation

Yang Feng, Yang Liu, Haitao Mi, Qun Liu, Yajuan Lü
Key Laboratory of Intelligent Information Processing
Institute of Computing Technology
Chinese Academy of Sciences
P.O. Box 2704, Beijing 100190, China
{fengyang, yliu, htmi, liuqun, lvyajuan}@ict.ac.cn

Abstract

Current system combination methods usually use confusion networks to find consensus translations among different systems. Requiring one-to-one mappings between the words in candidate translations, confusion networks have difficulty in handling more general situations in which several words are connected to another several words. Instead, we propose a lattice-based system combination model that allows for such phrase alignments and uses *lattice*s to encode all candidate translations. Experiments show that our approach achieves significant improvements over the state-of-the-art baseline system on Chinese-to-English translation test sets.

1 Introduction

System combination aims to find consensus translations among different machine translation systems. It has been proven that such consensus translations are usually better than the output of individual systems (Frederking and Nirenburg, 1994).

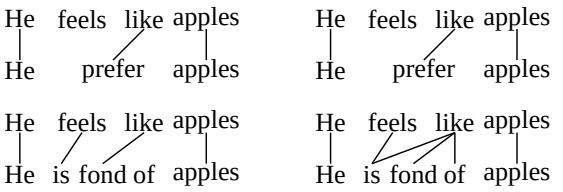
In recent several years, the system combination methods based on confusion networks developed rapidly (Bangalore et al., 2001; Matusov et al., 2006; Sim et al., 2007; Rosti et al., 2007a; Rosti et al., 2007b; Rosti et al., 2008; He et al., 2008), which show state-of-the-art performance in benchmarks. A confusion network consists of a sequence of sets of candidate words. Each candidate word is associated with a score. The optimal consensus translation can be obtained by selecting one word from each set to maximizing the overall score.

To construct a confusion network, one first need to choose one of the hypotheses (i.e., candidate translations) as the backbone (also called "skeleton" in the literature) and then decide the word alignments of other hypotheses to the backbone. Hypothesis alignment plays a crucial role in confusion-network-based system combination because it has a direct effect on selecting consensus translations.

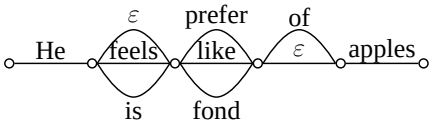
However, a confusion network is restricted in such a way that only 1-to-1 mappings are allowed in hypothesis alignment. This is not the fact even for word alignments between the same languages. It is more common that several words are connected to another several words. For example, "be capable of" and "be able to" have the same meaning. Although confusion-network-based approaches resort to inserting null words to alleviate this problem, they face the risk of producing degenerate translations such as "be capable to" and "be able of".

In this paper, we propose a new system combination method based on lattices. As a more general form of confusion network, a lattice is capable of describing arbitrary mappings in hypothesis alignment. In a lattice, each edge is associated with a sequence of words rather than a single word. Therefore, we select phrases instead of words in each candidate set and minimize the chance to produce unexpected translations such as "be capable to". We compared our approach with the state-of-the-art confusion-network-based system (He et al., 2008) and achieved a significant absolute improvement of 1.23 BLEU points on the NIST 2005 Chinese-to-English test set and 0.93 BLEU point on the NIST 2008 Chinese-to-English test set.

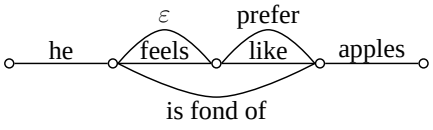
Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing, pages 1105–1113,
Singapore, 6-7 August 2009.



(a) unidirectional alignments (b) bidirectional alignments



(c) confusion network



(d) lattice

Figure 1: Comparison of a confusion network and a lattice.

2 Background

2.1 Confusion Network and Lattice

We use an example shown in Figure 1 to illustrate our idea. Suppose that there are three hypotheses:

He feels like apples
He prefer apples
He is fond of apples

We choose the first sentence as the backbone. Then, we perform hypothesis alignment to build a confusion network, as shown in Figure 1(a). Note that although "*feels like*" has the same meaning with "*is fond of*", a confusion network only allows for one-to-one mappings. In the confusion network shown in Figure 1(c), several null words ε are inserted to ensure that each hypothesis has the same length. As each edge in the confusion network only has a single word, it is possible to produce inappropriate translations such as "*He is like of apples*".

In contrast, we allow many-to-many mappings in the hypothesis alignment shown in Figure 2(b). For example, "*like*" is aligned to three words: "*is*", "*fond*", and "*of*". Then, we use a lattice shown in Figure 1(d) to represent all possible candidate translations. Note that the phrase "*is fond of*" is attached to an edge. Now, it is unlikely to obtain a translation like "*He is like of apples*".

A lattice $G = \langle V, E \rangle$ is a directed acyclic graph, formally a weighted finite state automation (FSA), where V is the set of nodes and E is the set of edges. The nodes in a lattice are usually labeled according to an appropriate numbering to reflect how to produce a translation. Each edge in a lattice is attached with a sequence of words as well as the associated probability.

As lattice is a more general form of confusion network (Dyer et al., 2008), we expect that replacing confusion networks with lattices will further improve system combination.

2.2 IHMM-based Alignment Method

Since the candidate hypotheses are aligned using Indirect-HMM-based (IHMM-based) alignment method (He et al., 2008) in both direction, we briefly review the IHMM-based alignment method first. Take the direction that the hypothesis is aligned to the backbone as an example. The conditional probability that the hypothesis is generated by the backbone is given by

$$p({e'_1}^J|e_1^I) = \sum_{a_1^J} \prod_{j=1}^{J} [p(a_j|a_{j-1}, I) p(e'_j|e_{a_j})] l \quad (1)$$

Where $e_1^I = (e_1, ..., e_I)$ is the backbone, ${e'}_1^J = (e'_1, ..., e'_J)$ is a hypothesis aligned to e_1^I, and $a_1^J = (a_1, .., a_J)$ is the alignment that specifies the position of backbone word that each hypothesis word is aligned to.

The translation probability $p(e'_j|e_i)$ is a linear interpolation of semantic similarity $p_{sem}(e'_j|e_i)$ and surface similarity $p_{sur}(e'_j|e_i)$ and α is the interpolation factor:

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

The semantic similarity model is derived by using the source word sequence as a hidden layer, so the bilingual dictionary is necessary. The semantic sim-

ilarity model is given by

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

The surface similarity model is estimated by calculating the literal matching rate:

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

where $s(e'_j, e_i)$ is given by

$$s(e'_j, e_i) = \frac{M(e'_j, e_i)}{\max(|e'_j|, |e_i|)} \quad (5)$$

where $M(e'_j, e_i)$ is the length of the longest matched prefix (LMP) and ρ is a smoothing factor that specifies the mapping.

The distortion probability $p(a_j = i|a_{j-1} = i', I)$ is estimated by only considering the jump distance:

$$p(i|i', I) = \frac{c(i - i')}{\sum_{i=1}^{I} c(l - i')} \quad (6)$$

The distortion parameters $c(d)$ are grouped into 11 buckets, $c(\leq -4)$, $c(-3)$, ..., $c(0)$, ..., $c(5)$, $c(\geq 6)$. Since the alignments are in the same language, the distortion model favor monotonic alignments and penalize non-monotonic alignments. It is given in a intuitive way

$$c(d) = (1 + |d - 1|)^{-K}, d = -4, ..., 6 \quad (7)$$

where K is tuned on held-out data.

Also the probability p_0 of jumping to a $null$ word state is tuned on held-out data. So the overall distortion model becomes

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

3 Lattice-based System Combination Model

Lattice-based system combination involves the following steps:

(1) Collect the hypotheses from the candidate systems.

(2) Choose the backbone from the hypotheses. This is performed using a sentence-level Minimum Bayes Risk (MBR) method. The hypothesis with the minimum cost of edits against all hypotheses is selected. The backbone is significant for it influences not only the word order, but also the following alignments. The backbone is selected as follows:

$$E_B = \underset{E' \in \mathbf{E}}{\operatorname{argmin}} \sum_{E \in \mathbf{E}} TER(E', E) \quad (8)$$

(3) Get the alignments of the backbone and hypothesis pairs. First, each pair is aligned in both directions using the IHMM-based alignment method. In the IHMM alignment model, bilingual dictionaries in both directions are indispensable. Then, we apply a grow-diag-final algorithm which is widely used in bilingual phrase extraction (Koehn et al., 2003) to monolingual alignments. The bidirectional alignments are combined to one resorting to the grow-diag-final algorithm, allowing n-to-n mappings.

(4)Normalize the alignment pairs. The word order of the backbone determines the word order of consensus outputs, so the word order of hypotheses must be consistent with that of the backbone. All words of a hypotheses are reordered according to the alignment to the backbone. For a word aligned to $null$, an actual $null$ word may be inserted to the proper position. The *alignment unit*s are extracted first and then the hypothesis words in each unit are shifted as a whole.

(5) Construct the lattice in the light of phrase pairs extracted on the normalized alignment pairs. The expression ability of the lattice depends on the phrase pairs.

(6) Decode the lattice using a model similar to the log-linear model.

The confusion-network-based system combination model goes in a similar way. The first two steps are the same as the lattice-based model. The difference is that the hypothesis pairs are aligned just in one direction due to the expression limit of the confusion network. As a result, the normalized alignments only contain 1-to-1 mappings (Actual $null$ words are also needed in the case of null alignment). In the following, we will give more details about the steps which are different in the two models.

4 Lattice Construction

Unlike a confusion network that operates words only, a lattice allows for phrase pairs. So phrase pairs must be extracted before constructing a lattice. A major difficulty in extracting phrase pairs is that the word order of hypotheses is not consistent with that of the backbone. As a result, hypothesis words belonging to a phrase pair may be discontinuous. Before phrase pairs are extracted, the hypothesis words should be normalized to make sure the words in a phrase pair is continuous. We call a phrase pair before normalization a *alignment unit*.

The problem mentioned above is shown in Figure 2. In Figure 2 (a), although ($e'_1e'_3$, e_2) should be a phrase pair, but "e'_1" and "e'_3" are discontinuous, so the phrase pair can not be extracted. Only after the words of the hypothesis are reordered according to the corresponding words in the backbone as shown in Figure 2 (b), "e'_1" and "e'_3" become continuous and the phrase pair ($e'_1e'_3$, e_2) can be extracted. The procedure of reordering is called *alignment normalization*

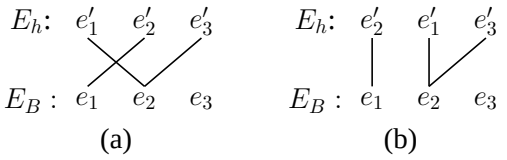


Figure 2: An example of alignment units

4.1 Alignment Normalization

After the final alignments are generated in the grow-diag-final algorithm, *minimum alignment unit*s are extracted. The hypothesis words of an alignment unit are packed as a whole in shift operations.

See the example in Figure 2 (a) first. All minimum alignment units are as follows: (e'_2, e_1), ($e'_1e'_3$, e_2) and (ε, e_3). ($e'_1e'_2e'_3$, e_1e_2) is an alignment unit, but not a minimum alignment unit.

Let $\bar{a}_i = (\bar{e}'_i, \bar{e}_i)$ denote a minimum alignment unit, and assume that the word string $\bar{e}'_i$ covers words e'_{i_1} ,..., e'_{i_m} on the hypothesis side, and the word string $\bar{e}_i$ covers the consecutive words e_{i_1} ,..., e_{i_n} on the backbone side. In an alignment unit, the word string on the hypothesis side can be discontinuous. The minimum unit $\bar{a}_i = (\bar{e}'_i, \bar{e}_i)$ must observe the following rules:

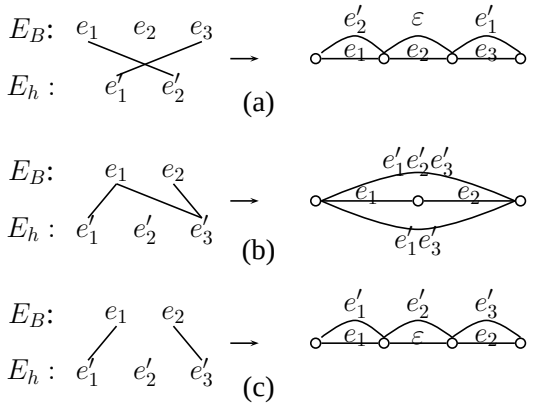


Figure 3: Different cases of $null$ insertion

- $\forall\ e'_{i_k} \in \bar{e}'_i, \quad e_{a'_{i_k}} \in \bar{e}_i$
- $\forall\ e_{i_k} \in \bar{e}_i, \quad e'_{a_{i_k}} = null$ or $e'_{a_{i_k}} \in \bar{e}'_i$
- $\nexists\ \bar{a}_j = (\bar{e}'_j, \bar{e}_j), \quad \bar{e}_j = e_{i_1}, ..., e_{i_k}$ or $\bar{e}_j = e_{i_k}, ..., e_{i_n}, \quad k \in [1, n]$

Where a'_{i_k} denotes the position of the word in the backbone that e'_{i_k} is aligned to, and a_{i_k} denotes the position of the word in the hypothesis that e_{i_k} is aligned to.

An actual $null$ word may be inserted to a proper position if a word, either from the hypothesis or from the backbone, is aligned to $null$. In this way, the minimum alignment set is extended to an alignment unit set, which includes not only minimum alignment units but also alignment units which are generated by adding $null$ words to minimum alignment units. In general, the following three conditions should be taken into consideration:

- A backbone word is aligned to $null$. A $null$ word is inserted to the hypothesis as shown in Figure 3 (a).
- A hypothesis word is aligned to $null$ and it is between the span of a minimum alignment unit. A new alignment unit is generated by inserting the hypothesis word aligned to null to the minimum alignment unit. The new hypothesis string must remain the original word order of the hypothesis. It is illustrated in Figure 3 (b).
- A hypothesis word is aligned to $null$ and it is not between the hypothesis span of any minimum alignment unit. In this case, a $null$ word

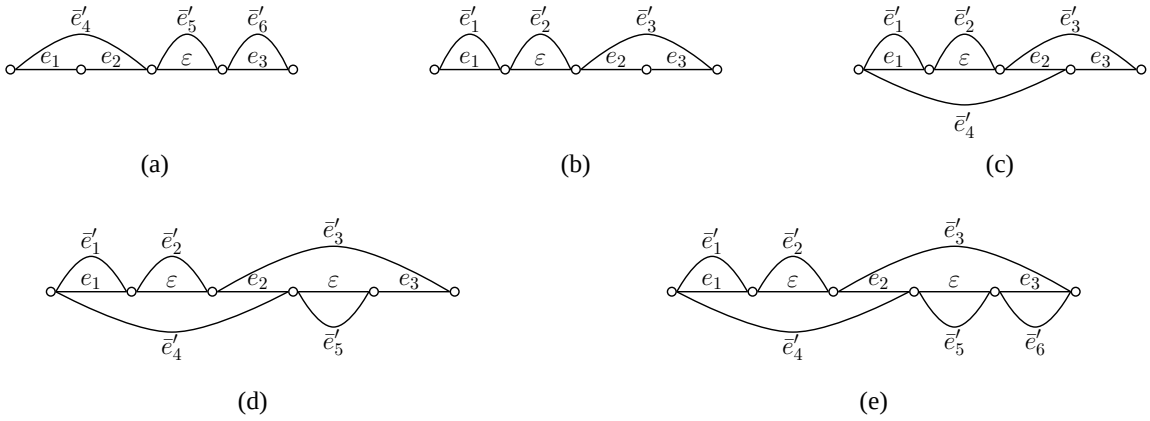


Figure 4: A toy instance of lattice construction

are inserted to the backbone. This is shown in Figure 3 (c).

4.2 Lattice Construction Algorithm

The lattice is constructed by adding the normalized alignment pairs incrementally. One backbone arc in a lattice can only span one backbone word. In contrast, all hypothesis words in an alignment unit must be packed into one hypothesis arc. First the lattice is initialized with a normalized alignment pair. Then given all other alignment pairs one by one, the lattice is modified dynamically by adding the hypothesis words of an alignment pair in a left-to-right fashion.

A toy instance is given in Figure 4 to illustrate the procedure of lattice construction. Assume the current inputs are: an alignment pair as in Figure 4 (a), and a lattice as in Figure 4 (b). The backbone words of the alignment pair are compared to the backbone words of the lattice one by one. The procedure is as follows:

- e_1 is compared with e_1. Since they are the same, the hypothesis arc $\bar{e}'_4$, which comes from the same node with e_1 in the alignment pair, is compared with the hypothesis arc $\bar{e}'_1$, which comes from the same node with e_1 in the lattice. The two hypothesis arcs are not the same, so $\bar{e}'_4$ is added to the lattice as shown in Figure 4(c). Both go to the next backbone words.

- e_2 is compared with ε. The lattice remains the same. The lattice goes to the next backbone word e_2.

- e_2 is compared with e_2. There is no hypothesis arc coming from the same node with the bone arc e_2 in the alignment pair, so the lattice remains the same. Both go to the next backbone words.

- ε is compared with e_3. A $null$ backbone arc is inserted into the lattice between e_2 and e_3. The hypothesis arc $\bar{e}'_5$ is inserted to the lattice, too. The modified lattice is shown in Figure 4(d). The alignment pair goes to the next backbone word e_3.

- e_3 is compared with e_3. For they are the same and there is no hypothesis arc $\bar{e}'_6$ in the lattice, $\bar{e}'_6$ is inserted to the lattice as in Figure 4(e).

- Both arrive at the end and it is the turn of the next alignment pair.

When comparing a backbone word of the given alignment pair with a backbone word of the lattice, the following three cases should be handled:

- The current backbone word of the given alignment pair is a $null$ word while the current backbone word of the lattice is not. A $null$ backbone word is inserted to the lattice.

- The current backbone word of the lattice is a $null$ word while the current word of the given alignment pair is not. The current $null$ backbone word of the lattice is skipped with nothing to do. The next backbone word of the lattice is compared with the current backbone word of the given alignment pair.

Algorithm 1 Lattice construction algorithm.

1: **Input**: alignment pairs $\{p_n\}_{n=1}^{N}$
2: $L \leftarrow p_1$
3: $Unique(L)$
4: **for** $n \leftarrow 2$ .. N **do**
5: $pnode = p_n \cdot first$
6: $lnode = L \cdot first$
7: **while** $pnode \cdot barcnext \neq NULL$ **do**
8: **if** $lnode \cdot barcnext = NULL$ or $pnode \cdot bword = null$ and $lnode \cdot bword \neq null$ **then**
9: INSERTBARC($lnode$, $null$)
10: $pnode = pnode \cdot barcnext$
11: **else**
12: **if** $pnode \cdot bword \neq null$ and $lnode \cdot bword = null$ **then**
13: $lnode = lnode \cdot barcnext$
14: **else**
15: **for** each $harc$ of $pnode$ **do**
16: **if** NotExist($lnode, pnode \cdot harc$) **then**
17: INSERTHARC($lnode$, $pnode \cdot harc$)
18: $pnode = pnode \cdot barcnext$
19: $lnode = lnode \cdot barcnext$
20: **Output:** lattice L

- The current backbone words of the given alignment pair and the lattice are the same. Let $\{harc_l\}$ denotes the set of hypothesis arcs, which come from the same node with the current backbone arc in the lattice, and $harc_h$ denotes one of the corresponding hypothesis arcs in the given alignment pair. In the $\{harc_l\}$, if there is no arc which is the same with the $harc_h$, a hypothesis arc projecting to $harc_h$ is added to the lattice.

The algorithm of constructing a lattice is illustrated in Algorithm 1. The backbone words of the alignment pair and the lattice are processed one by one in a left-to-right manner. Line 2 initializes the lattice with the first alignment pair, and Line 3 removes the hypothesis arc which contains the same words with the backbone arc. $barc$ denotes the backbone arc, storing one backbone word only, and $harc$ denotes the hypothesis arc, storing the hypothesis words. For there may be many alignment units span the same backbone word range, there may be more than one $harc$ coming from one node. Line $8-10$ consider the condition 1 and function *InsertBarc* in Line 9 inserts a $null$ bone arc to the position right before the current node. Line $12-13$ deal with condition 2 and jump to the next backbone word of the lattice. Line $15-19$ handle condition 3 and function *InsertHarc* inserts to the lattice a $harc$ with the same hypothesis words and the same backbone word span with the current hypothesis arc.

5 Decoding

In confusion network decoding, a translation is generated by traveling all the nodes from left to right. So a translation path contains all the nodes. While in lattice decoding, a translation path may skip some nodes as some hypothesis arcs may cross more than one backbone arc.

Similar to the features in Rosti et al. (2007a), the features adopted by lattice-based model are arc posterior probability, language model probability, the number of $null$ arcs, the number of hypothesis arcs possessing more than one non-null word and the number of all non-null words. The features are combined in a log-linear model with the arc posterior probabilities being processed specially as follows:

$$\begin{aligned}\log p(\mathbf{e}/\mathbf{f}) = &\sum_{i=1}^{N_{arc}} \log (\sum_{s=1}^{N_s} \lambda_s p_s(arc)) \\ &+ \zeta L(\mathbf{e}) + \alpha N_{nullarc}(\mathbf{e}) \\ &+ \beta N_{longarc}(\mathbf{e}) + \gamma N_{word}(\mathbf{e})\end{aligned} \tag{9}$$

where $\mathbf{f}$ denotes the source sentence, $\mathbf{e}$ denotes a translation generated by the lattice-based system, N_{arc} is the number of arcs the path of $\mathbf{e}$ covers, N_s is the number of candidate systems and λ_s is the weight of system s. ζ is the language model weight and $L(\mathbf{e})$ is the LM log-probability. $N_{nullarcs}(\mathbf{e})$ is the number of the arcs which only contain a $null$ word, and $N_{longarc}(\mathbf{e})$ is the number of the arcs which store more than one non-null word. The above two numbers are gotten by counting both backbone arcs and hypothesis arcs. α and β are the corresponding weights of the numbers, respectively. $N_{word}(\mathbf{e})$ is the non-null word number and γ is its weight.

Each arc has different confidences concerned with different systems, and the confidence of system s is denoted by $p_s(arc)$. $p_s(arc)$ is increased by

$1/(k+1)$ if the hypothesis ranking k in the system s contains the arc (Rosti et al., 2007a; He et al., 2008).

Cube pruning algorithm with beam search is employed to search for the consensus output (Huang and Chiang, 2005). The nodes in the lattice are searched in a topological order and each node retains a list of N best candidate partial translations.

6 Experiments

The candidate systems participating in the system combination are as listed in Table 1: System A is a BTG-based system using a MaxEnt-based reordering model; System B is a hierarchical phrase-based system; System C is the Moses decoder (Koehn et al., 2007); System D is a syntax-based system. 10-best hypotheses from each candidate system on the dev and test sets were collected as the input of the system combination.

In our experiments, the weights were all tuned on the NIST MT02 Chinese-to-English test set, including 878 sentences, and the test data was the NIST MT05 Chinese-to-English test set, including 1082 sentences, except the experiments in Table 2. A 5-gram language model was used which was trained on the XinHua portion of Gigaword corpus. The results were all reported in case sensitive BLEU score and the weights were tuned in Powell's method to maximum BLEU score. The IHMM-based alignment module was implemented according to He et al. (2008), He (2007) and Vogel et al. (1996). In all experiments, the parameters for IHMM-based alignment module were set to: the smoothing factor for the surface similarity model, $\rho = 3$; the controlling factor for the distortion model, $K = 2$.

6.1 Comparison with Confusion-network-based model

In order to compare the lattice-based system with the confusion-network-based system fairly, we used IHMM-based system combination model on behalf of the confusion-network-based model described in He et al. (2008). In both lattice-based and IHMM-based systems, the bilingual dictionaries were extracted on the FBIS data set which included 289K sentence pairs. The interpolation factor of the similarity model was set to $\alpha = 0.1$.

The results are shown in Table 1. *IHMM* stands for the IHMM-based model and *Lattice* stands for the lattice-based model. On the dev set, the lattice-based system was 3.92 BLEU points higher than the best single system and 0.36 BLEU point higher than the IHMM-based system. On the test set, the lattice-based system got an absolute improvement by 3.73 BLEU points over the best single system and 1.23 BLEU points over the IHMM-based system.

System	MT02 BLEU%	MT05 BLEU%
SystemA	31.93	30.68
SystemB	32.16	32.07
SystemC	32.09	31.64
SystemD	33.37	31.26
IHMM	36.93	34.57
Lattice	37.29	35.80

Table 1: Results on the MT02 and MT05 test sets

The results on another test sets are reported in Table 2. The parameters were tuned on the newswire part of NIST MT06 Chinese-to-English test set, including 616 sentences, and the test set was NIST MT08 Chinese-to-English test set, including 1357 sentences. The BLEU score of the lattice-based system is 0.93 BLEU point higher than the IHMM-based system and 3.0 BLEU points higher than the best single system.

System	MT06 BLEU%	MT08 BLEU%
SystemA	32.51	25.63
SystemB	31.43	26.32
SystemC	31.50	23.43
SystemD	32.41	26.28
IHMM	36.05	28.39
Lattice	36.53	29.32

Table 2: Results on the MT06 and MT08 test sets

We take a real example from the output of the two systems (in Table 3) to show that higher BLEU scores correspond to better alignments and better translations. The translation of System C is selected as the backbone. From Table 3, we can see that because of 1-to-1 mappings, "*Russia*" is aligned to "*Russian*" and "*'s*" to "*null*" in the IHMM-based model, which leads to the error translation "*Russian*

Source:	俄罗斯 国营 石油 公司 正 与 俄罗斯 国营 瓦斯 公司 进行合并
SystemA:	Russia merger of state-owned oil company and the state-run gas company in Russia
SystemB:	Russia 's state-owned oil company is working with Russia 's state-run gas company mergers
SystemC:	Russian state-run oil company is combined with the Russian state-run gas company
SystemD:	Russia 's state-owned oil companies are combined with Russia 's state-run gas company
IHMM:	Russian 's state-owned oil company working with Russia 's state-run gas company
Lattice:	Russia 's state-owned oil company is combined with the Russian state-run gas company

Table 3: A real translation example

's". Instead, "*Russia 's*" is together aligned to "*Russian*" in the lattice-based model. Also due to 1-to-1 mappings, $null$ word aligned to "*is*" is inserted. As a result, "*is*" is missed in the output of IHMM-based model. In contrast, in the lattice-based system, "*is working with*" are aligned to "*is combined with*", forming a phrase pair.

6.2 Effect of Dictionary Scale

The dictionary is important to the semantic similarity model in IHMM-based alignment method. We evaluated the effect of the dictionary scale by using dictionaries extracted on different data sets. The dictionaries were respectively extracted on similar data sets: 30K sentence pairs, 60K sentence pairs, 289K sentence pairs (FBIS corpus) and 2500K sentence pairs. The results are illustrated in Table 4. In order to demonstrate the effect of the dictionary size clearly, the interpolation factor of similarity model was all set to $\alpha = 0.1$.

From Table 4, we can see that when the corpus size rise from 30k to 60k, the improvements were not obvious both on the dev set and on the test set. As the corpus was expanded to 289K, although on the dev set, the result was only 0.2 BLEU point higher, on the test set, it was 0.63 BLEU point higher. As the corpus size was up to 2500K, the BLEU scores both on the dev and test sets declined. The reason is that, on one hand, there are more noise on the 2500K sentence pairs; on the other hand, the 289K sentence pairs cover most of the words appearing on the test set. So we can conclude that in order to get better results, the dictionary scale must be up to some certain scale. If the dictionary is much smaller, the result will be impacted dramatically.

	MT02 BLEU%	MT05 BLEU%
30k	36.94	35.14
60k	37.09	35.17
289k	37.29	35.80
2500k	37.14	35.62

Table 4: Effect of dictionary scale

6.3 Effect of Semantic Alignments

For the IHMM-based alignment method, the translation probability of an English word pair is computed using a linear interpolation of the semantic similarity and the surface similarity. So the two similarity models decide the translation probability together and the proportion is controlled by the interpolation factor. We evaluated the effect of the two similarity models by varying the interpolation factor α.

We used the dictionaries extracted on the FBIS data set. The result is shown in Table 5. We got the best result with $\alpha = 0.1$. When we excluded the semantic similarity model ($\alpha = 0.0$) or excluded the surface similarity model ($\alpha = 1.0$), the performance became worse.

7 Conclusion

The alignment model plays an important role in system combination. Because of the expression limitation of confusion networks, only 1-to-1 mappings are employed in the confusion-network-based model. This paper proposes a lattice-based system combination model. As a general form of confusion networks, lattices can express n-to-n mappings. So a lattice-based model processes phrase pairs while

	MT02 BLEU%	MT05 BLEU%
$\alpha = 1.0$	36.41	34.92
$\alpha = 0.7$	37.21	35.65
$\alpha = 0.5$	36.43	35.02
$\alpha = 0.4$	37.14	35.55
$\alpha = 0.3$	36.75	35.66
$\alpha = 0.2$	36.81	35.55
$\alpha = 0.1$	37.29	35.80
$\alpha = 0.0$	36.45	35.14

Table 5: Effect of semantic alignments

a confusion-network-based model processes words only. As a result, phrase pairs must be extracted before constructing a lattice.

On NIST MT05 test set, the lattice-based system gave better results with an absolute improvement of 1.23 BLEU points over the confusion-network-based system (He et al., 2008) and 3.73 BLEU points over the best single system. On NIST MT08 test set, the lattice-based system outperformed the confusion-network-based system by 0.93 BLEU point and outperformed the best single system by 3.0 BLEU points.

8 Acknowledgement

The authors were supported by National Natural Science Foundation of China Contract 60736014, National Natural Science Foundation of China Contract 60873167 and High Technology R&D Program Project No. 2006AA010108. Thank Wenbin Jiang, Tian Xia and Shu Cai for their help. We are also grateful to the anonymous reviewers for their valuable comments.

References

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

Christopher Dyer, Smaranda Muresan, and Philip Resnik. 2008. Generalizing word lattice translation. In *Proceedings of ACL/HLT 2008*, pages 1012–1020, Columbus, Ohio, June.

Robert Frederking and Sergei Nirenburg. 1994. Three heads are better than one. In *Proc. of ANLP*, pages 95–100.

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

Xiaodong He. 2007. Using word-dependent translation models in hmm based word alignment for statistical machine translation. In *Proc. of COLING-ACL*, pages 961–968.

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

Philipp Koehn, Franz J. Och, and Daniel Marcu. 2003. Statistical phrase-based translation. In *Proc. of HLT-NAACL*, pages 127–133.

Philipp Koehn, Hieu Hoang, Alexandra Birch Mayne, Christopher Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondrej Bojar, Alexandra Constantin, and Evan Herbst. 2007. Moses: Open source toolkit for statistical machine translation. In *Proc. of the 45th ACL, Demonstration Session*.

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

Antti-Veikko I. Rosti, Spyros Matsoukas, and Richard Schwartz. 2007a. Improved word-level system combination for machine translation. In *Proc. of ACL*, pages 312–319.

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

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

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.

Stephan Vogel, Hermann Ney, and Christoph Tillmann. 1996. Hmm-based word alignment in statistical translation. In *Proc. of COLING*, pages 836–841.