A Source-side Decoding Sequence Model for Statistical Machine Translation

Minwei Feng and **Arne Mauser** and **Hermann Ney**
Human Language Technology and Pattern Recognition Group
RWTH Aachen University, Germany
<surname>@cs.rwth-aachen.de

Abstract

We propose a source-side decoding sequence language model for phrase-based statistical machine translation. This model is a reordering model in the sense that it helps the decoder find the correct decoding sequence. The model uses word-aligned bilingual training data. We show improved translation quality of up to 1.34% BLEU and 0.54% TER using this model compared to three other widely used reordering models.

1 Introduction

The systematic word order difference between two languages, pose a challenge for current statistical machine translation (SMT) systems. The system has to decide in which order to translate the given source words. This problem is known as the reordering problem. As shown in (Knight, 1999), if arbitrary reordering is allowed, the search problem is NP-hard.

Many ideas have been proposed to address the reordering problem. Within the phrase-based SMT framework there are mainly three stages where improved reordering could be integrated:

1. Reorder the source sentence. So that the word order of source and target sentences is similar. Usually it is done as the preprocessing step for both training data and test data.

2. In the decoder, add models in the log-linear framework or constraints in the decoder to reward good reordering options or penalize bad ones.

3. In the reranking framework.

For the first point, (Wang et al., 2007) used manually designed rules to reorder parse trees of the source sentences as a preprocessing step. Based on shallow syntax, (Zhang et al., 2007) used rules to reorder the source sentences on the chunk level and provide a source-reordering lattice instead of a single reordered source sentence as input to the SMT system. Designing rules to reorder the source sentence is conceptually clear and usually easy to implement. In this way, syntax information can be incorporated into phrase-based SMT systems. However, one disadvantage is that the reliability of the rules is often language pair dependent. In practice, another problem is that the experiments are often time consuming. Once the preprocess has been changed, all later steps like alignment training and translation model training must be redone.

In the second category, researchers try to inform the decoder on what a good reordering is or what a suitable decoding sequence is. (Zens and Ney, 2006) used a discriminative reordering model to predict the orientation of the next phrase given the previous phrase. (Chang et al., 2009) used the same idea as above but additionally they adopt path features extracted from dependency trees. (Cherry, 2008) put the syntactic cohesion as a soft constraint in the decoder to guide the decoding process to choose those translations that do dot violate the syntactic structure of the source sentence. Since the decoder uses a log-linear framework, the new feature can be easily added. Another advantage of methods in this category is that we let the decoder decide the weights of features, so that even if one model gives wrong estimation sometimes, it can still be corrected by other models. Our work in this paper belongs to this category.

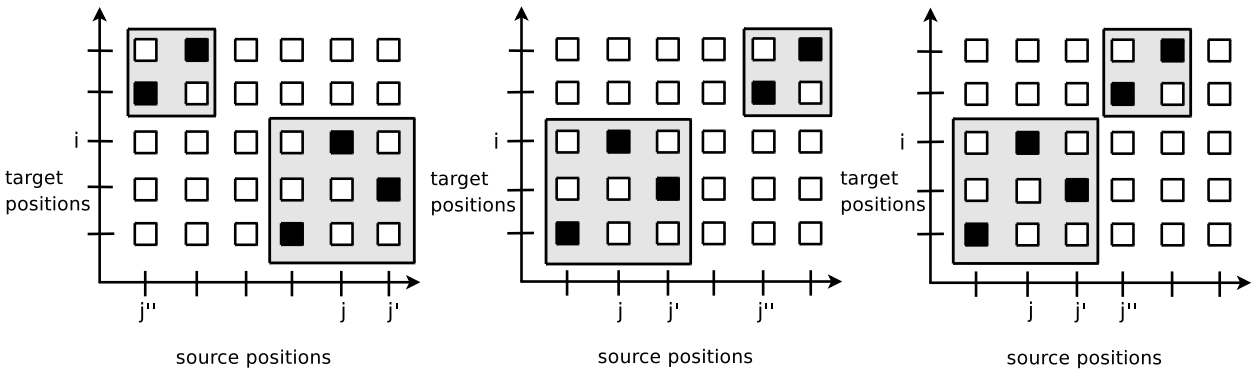

Figure 1: Phrase orientation: left, right and monotone. j is the source word position aligned to the last target word of current phrase. $j^{'}$ is the last source word position of current phrase. $j^{''}$ is the source word position aligned to the first target word position of the next phrase.

In the reranking step, the system has the last opportunity to choose a good translation. (Och et al., 2004) describe the use of syntactic features in the rescoring step. They report the most useful feature is IBM Model 1 score. The syntactic features contribute very small gains. Another disadvantage of carrying out reordering in reranking is the representativeness of the N-best list is often a question mark.

In this paper, we propose a source-side decoding sequence language model (LM) which can help the decoder to find an appropriate source word translation order. Three steps need to be done: (1) alignment training; (2) extract the reordered source corpus from the alignment; (3) train the LM on the reordered corpus. This LM is then added into the decoder as one feature of the log-linear framework.

The rest of the paper is organized as follows. Section 2 reviews the related work. In Section 3 we summarize the baseline system used for experiments. The proposed model will be described in Section 4 . The experimental setting and translation results are presented in Section 5 and Section 6. We will make detailed analysis in Section 7 and finally draw the conclusions in Section 8.

2 Related Work

Many approaches to reordering in machine translation have been proposed in the past. An approach that is closely related to the one presented here is (Costa-jussà and Fonollosa, 2008) where reordering lattices are generated using an SMT system and then weighted using a word-class n-gram language model trained on reordered data. This reordered data is obtained from word-aligned bilingual training data. While our approach is similar in that we learn an n-gram model on reordered source data, we use full form words and do not rely on permutation graphs.

As we compare our method with (Zens and Ney, 2006) and (Cherry, 2008), we will give a short summary of those two models here.

We use Figure 1 as an example. j is the source word position which is aligned to the last target word of the current phrase. $j^{'}$ is the last source word position of the current phrase. $j^{''}$ is the source word position which is aligned to the first target word position of the next phrase. (Zens and Ney, 2006) proposed a maximum entropy classifier to predict the orientation of the next phrase given the current phrase. The orientation class $c_{j,j^{'},j^{''}}$ is defined as:

$$c_{j,j^{'},j^{''}} = \begin{cases} \text{left}, & \text{if } j^{''} < j \\ \text{right}, & \text{if } j^{''} > j \text{ and } j^{''} - j^{'} > 1 \\ \text{monotone}, & \text{if } j^{''} > j \text{ and } j^{''} - j^{'} = 1 \end{cases} \quad (1)$$

The orientation probability is modeled in a log-linear framework using a set of N feature functions $h_n(f_1^J, e_1^I, i, j, c_{j,j^{'},j^{''}})$, $n = 1, \ldots, N$. The whole model is:

$$p_{\lambda_1^N}(c_{j,j^{'},j^{''}} | f_1^J, e_1^I, i, j) = \frac{\exp(\sum_{n=1}^{N} \lambda_n h_n(f_1^J, e_1^I, i, j, c_{j,j^{'},j^{''}}))}{\sum_{c^{'}} \exp(\sum_{n=1}^{N} \lambda_n h_n(f_1^J, e_1^I, i, j, c^{'}))} \quad (2)$$

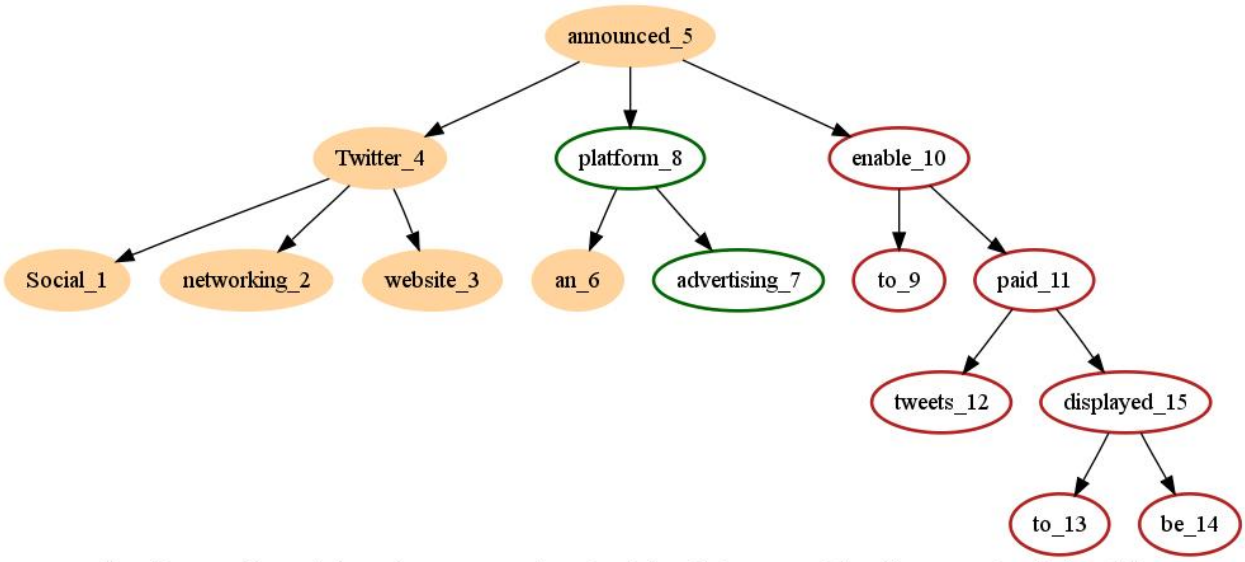

Figure 2: A dependency tree. The source sentence is given at bottom. Every node represents a source word.

Different features can be used, e.g. source/target words within a window, word classes or part-of-speech within a window around the current source/target position. We use the source word features to train the model. As shown in (Zens and Ney, 2006), the source word features give the main contribution.

We now describe another model which is used for comparison later. (Cherry, 2008) proposed a syntactic cohesion model. The core idea is that the syntactic structure of the source sentence should be preserved in translation. This structure is chosen to be represented by a dependency tree. To keep syntactic cohesion, the decoding process should not break this dependency structure. (Cherry, 2008) used his model as a new feature of the log-linear decoding framework and showed improvement on English-to-French direction. We implement this model in the phrase-based decoder and report results on Chinese-to-English translation.

To illustrate the method, we use the Figure 2 as an example. Figure 2 is a dependency tree (suppose the source sentence is now English). Every node represents a word. We also put the position of the word in the node. So *Social_1* means *Social* is the first word of the sentence. The algorithm is as follows.

Given the source sentence and its dependency tree, during the translation process, once a hypothesis is extended, check if there exists a subtree T such that:

- Its translation is already started (at least one node is covered)
- It is interrupted by the new added phrase (at least one word in the new source phrase is not in T)
- It is not finished (after the new phrase is added, there is still at least one free node in T)

If so, we say this hypothesis violates the subtree T, and the model returns the number of subtrees that this hypothesis violates.

In Figure 2, nodes filled with yellow means the words are already translated. Now suppose the length of the new added phrase is one, then according to the above algorithm only position 7 and 8 (green ellipse) are good candidates. Choosing other source words (red ellipse) to translate will violate the subtree *an_6—advertising_7—platform_8*.

3 Translation System Overview

In this section, we are going to describe the phrase-based SMT system we used for the experiments. In statistical machine translation, we are given a source

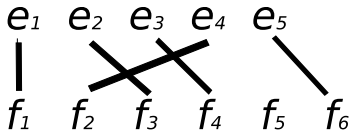

Figure 3: Original alignment

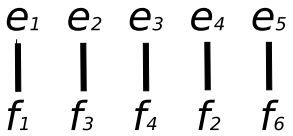

Figure 4: After reordering

Figure 5: From source sentence to decoding sequence

language sentence $f_1^J = f_1 \dots f_j \dots f_J$. The objective is to translate the source into a target language sentence $e_1^I = e_1 \dots e_i \dots e_I$. The strategy is among all possible target language sentences, we will choose the one with the highest probability:

$$\hat{e}_i^{\hat{I}} = \arg\max_{I, e_1^I} \{Pr(e_1^I | f_1^J)\} \quad (3)$$

We model $Pr(e_1^I | f_1^J)$ directly using a log-linear combination of several models (Och and Ney, 2002):

$$Pr(e_1^I | f_1^J) = \frac{\exp\Big(\sum_{m=1}^{M} \lambda_m h_m(e_1^I, f_1^J)\Big)}{\sum_{I', e'^{I'}_1} \exp\Big(\sum_{m=1}^{M} \lambda_m h_m(e'^{I'}_1, f_1^J)\Big)} \quad (4)$$

The denominator is to make the $Pr(e_1^I | f_1^J)$ to be a probability distribution and it depends only on the source sentence f_1^J. For search, the decision rule is simply:

$$\hat{e}_i^{\hat{I}} = \arg\max \Big\{ \sum_{m=1}^{M} \lambda_m h_m(e_1^I, f_1^J) \Big\} \quad (5)$$

The model scaling factors λ_1^M are trained with Minimum Error Rate Training (MERT).

Our baseline is a state-of-art phrase-based translation system (Zens, 2008). The baseline includes the following models: an n-gram target-side language model, a phrase translation model and a word-based lexicon model. The latter two models are used for both directions: $p(f|e)$ and $p(e|f)$. Additionally we use phrase count features, word and phrase penalty. The reordering model for the baseline system is a simple distance-based model which assigns costs based on distance from the end of previous phrase to the start of the current phrase.

4 Source-side Decoding Sequence LM

In this section, we will introduce the proposed model. First we will describe the training process. Then we explain how to use it in the decoder.

4.1 Modeling

The language model plays a crucial role for translation performance. With mature smoothing algorithms like (Kneser and Ney, 1995) and open-source toolkits like (Stolcke, 2002), the LM provides a simple and efficient method to use large amount of data. For SMT, the conventional target language LM is used as one feature in Equation (4).

Figure 3 shows an alignment after GIZA++ training. If we regard this alignment as a translation result, i.e. given the source sentence f_1^6, the system translates it into the target sentence e_1^5. The alignment link set $\{a_1 = 1, a_2 = 4, a_3 = 2, a_4 = 3, a_5 = 0, a_6 = 5\}$ reveals the decoding process, i.e. the alignment implies the order in which the source words should be translated, e.g. after f_1 is translated we have e_1 on the target side. e_2 is linked to f_3 which tells us f_3 is the second source word to be translated. We reorder the source side of the alignment to get Figure 4. The source side of this monotone alignment is then extracted. This process is done for every sentence pair of the bilingual corpora after the alignment training is finished. So we transfer the source corpora to the source decoding order corpora (Figure 5) and use the corpora to train a LM. The reader may notice there is an unaligned word f_5 here. What we do here is to simply ignore those unaligned words. The reason is that we regard the alignment as a source decoding sequence, so an unaligned word means it does not need to be translated. One can also deal with it by attaching the

unaligned word to some designed positions, e.g. after its predecessor. In Figure 6, we explained how the decoding order will be extracted for two cases. Figure 6 contains an example of m-to-1 alignment and an example of 1-to-m alignment. The source decoding sequence extracted from this alignment is $f_1 f_5 f_2 f_3 f_4 f_5$. Namely, when m source words are linked to the same target word, their original order is kept. When one source word is linked to multi contiguous target words (f_5 is linked to e_5, e_6), it is regarded as appearing once. If one source word is linked to multi separate words (f_5 is linked to e_2, e_5), then the source word will be repeated in the decoding sequence.

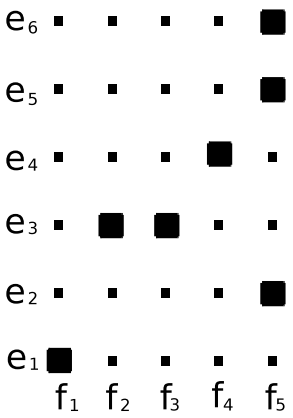

Figure 6: Source decoding sequence extraction for m-to-1 and 1-to-m alignment cases. The source decoding sequence extracted from this alignment is $f_1 f_5 f_2 f_3 f_4 f_5$.

4.2 Decoding

Like the two methods we described in Section 2, we also add this source-side decoding sequence LM as a new feature in Equation (4). The usage of our model is similar to the standard target-side LM.

The states in the standard search space are identified by the triple (C, h_e, j). C is the coverage vector indicating those source words that have been translated. h_e is the target-side language model history and j is the last source word position in current source phrase $\tilde{f}$. To use our model, the search state must be augmented with the source-side decoding sequence LM history h_f. So the search state now is a quadruple (C, h_e, h_f, j).

In search, we can apply the source-side LM directly when scoring an extended state. Suppose the search state is now extended with a new phrase pair $(\tilde{f}, \tilde{e})$. $\tilde{F}$ is the extracted source decoding sequence for the new phrase pair $(\tilde{f}, \tilde{e})$ and $\tilde{F}^i$ is the i^{th} word within $\tilde{F}$. $\tilde{F}'$ is the source-side decoding sequence history for current state. We compute the feature score $h_{wolm}(\tilde{F}, \tilde{F}')$ of the extended state as follows.

$$h_{wolm}(\tilde{F}, \tilde{F}') = \lambda \cdot \sum_{i=1}^{|\tilde{F}|} \log p(\tilde{F}^i | \tilde{F}', \tilde{F}^1, \cdots, \tilde{F}^{i-1}) \tag{6}$$

λ is the scaling factor for this model. $|\tilde{F}|$ is the length of this decoding sequence. For a sentence pair (f_1^J, e_1^I) and its source decoding sequence F, our model returns the following probability value:

$$p(e_1^I | f_1^J) = p(F) = \prod_{i=1}^{|F|} p(F^i | F^1, \cdots, F^{i-1}) \tag{7}$$

$|F|$ is the length of decoding sequence F. Now only given the sentence pair (f_1^J, e_1^I), we sum all possible decoding sequences probability:

$$\sum_{\forall F} \prod_{i=1}^{|F|} p(F^i | F^1, \cdots, F^{i-1}) \tag{8}$$

This value is usually not equal to one, as all $p(F^i | F^1, \cdots, F^{i-1})$ are from the LM which is trained on the whole corpora. However, our purpose is to do translation by the search Equation (5). The most important function of the model is that it can distinguish between good and bad decoding sequences.

The way that the model scores the hypothesis during search must be consistent with the extraction process in the training step. We give an example here. Suppose the source decoding history for current search state is a, b. The state is extended with the new phrase pair in Figure 6. As we mentioned in previous subsection, the extracted decoding sequence will be $f_1 f_5 f_2 f_3 f_4 f_5$. This new model will then return the following score for this extended

state:

$$
\begin{aligned}
&h_{wolm}(\tilde{F}, \tilde{F}') \\
&= \lambda \cdot \big[\log p(f_1|a,b) + \log p(f_5|a,b,f_1) \\
&\quad + \log p(f_2|a,b,f_1,f_5) + \log p(\tilde{f}_3|a,b,f_1,f_5,f_2) \\
&\quad + \log p(f_4|a,b,f_1,f_5,f_2,f_3) \\
&\quad + \log p(\tilde{f}_5|a,b,f_1,f_5,f_2,f_3,f_4) \big]
\end{aligned} \tag{9}
$$

Compared to the usage of the target-side LM, the loading time of this source-side decoding sequence LM is much shorter. We only need to load the n-grams that will be used during the search, i.e., those n-gram items consists of the words in the specific sentence. We already have the source sentence before search. So only small number of n-grams will be loaded into the decoder. In principle, the decoding time and the memory consumption will be increased by adding this model. However, the amount is so small that we do not feel this resource consumption increase.

5 Experimental Evaluation

5.1 Statistics

The experiments were conducted on the NIST Chinese-to-English translation task. We use the NIST06 test file and its references as tuning data. Results for test files NIST02, NIST03, NIST04, NIST05 and NIST08 are given. Table 1 shows the corpus statistics.

The language models for source and target are both trained by SRILM toolkit (Stolcke, 2002) with modified Kneser-Ney smoothing. The 9-gram source-side LM is trained on the reordered source sentences of the bilingual data in Table 1 (source sentences are filtered out if the length is ≤ 3). The 6-gram target-side LM is trained on the English-side of the whole bilingual corpus and additional monolingual English data from GigaWordV3 (LDC2007T07). Table 2 shows the LM statistics. We hope our source-side decoding sequence LM can catch some global reordering by setting a high order of the LM. However, as we can see from Table 2, due to limited amount of training data, the number of n-gram starts to drop from 4-gram. At the same time, the number of n-gram on target-side LM keeps increasing because we use huge amount of additional data from GigaWord V3. So the sparseness problem is much less severe than the source-side LM.

		Chinese	English
Train:	Sentences	8M	
	Running Words	223M	238M
	Vocabulary	166K	365K
Dev(06):	Sentences	1664	
	Running Words	41K	188K
Test(02):	Sentences	878	
	Running Words	25K	105K
Test(03):	Sentences	919	
	Running Words	26K	122K
Test(04):	Sentences	1788	
	Running Words	52K	245K
Test(05):	Sentences	1082	
	Running Words	33K	148K
Test(08):	Sentences	1357	
	Running Words	35K	165K

Table 1: Chinese-English NIST task: corpus statistics for training, tuning and test data. For tuning and test data, each source sentence has four references.

However, we still hope that the decoder can benefit from those high order n-grams. So we used this 9-gram source-side LM. For the source decoding sequence LM, we selected some sentence pairs from LDC2006E93 corpus which has manually annotated word alignment. However, the alignment for Chinese-English is made on character level. So we convert it to the word level alignment and get the reordered source sentences as a test file to calculate the perplexity (PPL). Perplexity is given in Table 3. As we see from the Table 3, the perplexity value 321.628 is a relatively high score according to our

n-gram order	Source LM	Target LM
1	163K	1104K
2	15738K	51795K
3	16899K	96813K
4	16268K	177951K
5	13709K	217131K
6	11986K	235943K
7	10834K	-
8	9980K	-
9	9959K	-

Table 2: source 9-gram LM and target 6-gram LM statistics

LM	Test sentences	Test running words	Test OOVs(running words)	PPL
Src decoding sequence LM	10304	235K	57	321.628

Table 3: Test file statistics and perplexity(PPL) for the source-side decoding sequence LM

experiences. We believe the main reasons are the following:

1. Training data amount. As we mentioned before, unlike target-side LM which includes huge amount data of GigaWord V3, the source-side LM is only trained on bilingual corpora. The file size of our source-side decoding sequence LM is 1.8G while the file size of target-side LM is 12G.

2. Training data quality. Different with target-side training data which is English text, the source-side LM is trained on a corpora extracted from the word alignment, which contains many errors.

3. The test file quality. The PPL is a measure of distance between the training data probability distribution and test data probability distribution. The test file for the source-side LM is a human-annotated alignment file, which is quite different from the machine generated alignment. For example, the human made corpora contains many m-to-n alignment which will not appear in the machine generated alignment. We convert the m-to-n alignment to 1-to-1 before calculate the PPL. However, this might not be the best way to do it.

In spite of the above disadvantages, the decoder can still utilize useful information from the model and provides better translation results.

5.2 Reordered Source Data

To get some feeling about to what extent the source sentences have been reordered during the extraction process in Figure 5, we calculated the Word Error Rate (WER) and Position-independent Error Rate (PER). WER (Nießen et al., 2000) is based on Levenshtein distance and normalized by the reference length. PER (Tillmann et al., 1997) ignores the positions and calculates Insertions/Deletions/Substitutions on bag-of-words and then normalized by the reference length. The difference between WER and PER gives a hint how much words are reordered. The results are in Table 4.

WER [%]	PER [%]	WER - PER [%]
57.08	12.14	44.94

Table 4: Error rate between the original source corpus and the reordered source corpus

6 Translation Results

In this section, we report translation results on the NIST task using the automatic evaluation measures BLEU (Papineni et al., 2001) and TER (Snover et al., 2005).

In Table 5, we show the results on different test corpora. The baseline system **base** includes a distance model. System **base+mero** adds (Zens and Ney, 2006)'s model. **mero** means maximum entropy reordering model. **base+sc** is the baseline system plus (Cherry, 2008)'s model. **sc** is the abbreviation of syntactic cohesion. We use the Stanford Parser (Levy and Manning, 2003) to get the Chinese dependency trees. **base+wolm** is the baseline system plus the source-side decoding sequence LM model. **wolm** means word order language model.

In Table 5, first four lines are the comparison of single reordering models. **Bold** number indicates the best system in that group. The results show that all three models give improvements compared to the **base** system. For BLEU metric, out of five test files, **base+wolm** wins three cases and **base+mero** wins the rest two. For TER metric, **base+sc** always plays the first place. The last two lines are the comparison of the total behavior of the models. We see **base+sc+mero+wolm** always performs better than **base+sc+mero**.

Some translation examples are shown in Table 6. The quality of different translation is basically consistent with the scores in Table 5. Both three models improve the translation quality. Translations with **sc** and **wolm** are more understandable.

HYP (BLEU[%])	NIST02	NIST03	NIST04	NIST05	NIST08
base	34.79	34.77	35.09	33.28	24.25
base+mero	35.28	34.82	**36.05**	34.43	**26.31**
base+sc	35.32	35.21	35.67	34.53	25.92
base+wolm	**35.37**	**35.37**	35.86	**34.61**	25.59
base+sc+mero	36.35	36.36	36.93	35.66	26.09
base+sc+mero+wolm	**36.38**	**36.71**	**37.14**	**35.68**	**26.47**
HYP (TER[%])	NIST02	NIST03	NIST04	NIST05	NIST08
base	59.23	59.40	59.54	59.63	66.29
base+mero	59.73	60.01	59.19	59.57	65.73
base+sc	**59.04**	**58.82**	**58.79**	**58.68**	**65.23**
base+wolm	59.36	59.16	59.00	59.35	65.85
base+sc+mero	59.03	58.22	58.08	58.04	65.13
base+sc+mero+wolm	**58.37**	**57.90**	**57.72**	**57.90**	**64.76**

Table 5: Translation results for several evaluation sets of Chinese-English NIST task. Scores are calculated in case-insensitive way.

base+sc+mero+wolm performs a little bit better than **base+sc+mero**.

7 Discussion

We compared our proposed model **wolm** to three different reordering models, namely, (Zens and Ney, 2006)'s **mero**, (Cherry, 2008)'s **sc** and the simple distance-based model **base**. All those models have the same objective: to guide the decoder to have a good translation order. **sc** only uses the source-side structure information, while **wolm** is trained on the reordered source sentences and **mero** uses source words features. In order to have the training data, both **wolm** and **mero** use the alignment which is trained on the bilingual corpus. All three models (**mero**, **sc**, **wolm**) have the same disadvantage: they all rely on the information that includes errors. **sc** takes a dependency tree generated by a parser which is definitely not perfect. The alignment utilized by **wolm** and **mero** also contains lots of errors.

Although facing the above problems, these three models are all able to improve the translation quality compared to the distance-based model **base**. The first group in Table 5 shows the single model comparison. Compared to **base**, **sc** gives 0.44% to 1.67% BLEU improvement, **mero** provides 0.05% to 2.06% BLEU improvements and our **wolm** contributes 0.58% to 1.34% BLEU improvements. For TER scores, **sc** decreases the value by 0.19% to 1.06%, **mero** reduces the TER by -0.61% to 0.56% and **wolm** cuts down the error by -0.13% to 0.54%. We see some fluctuations here. For **mero** and **wolm**, the TER value gets worse sometimes. The second group in Table 5 is to see if **wolm** can improve the system **base+sc+mero**. For BLEU, the full system **base+sc+mero+wolm** is 0.03% to 0.38% higher the **base+sc+mero** system and TER is reduced by 0.14% to 0.66%. The results are much more consistent than the single model comparison cases.

8 Conclusion

In this paper we proposed a source-side decoding sequence LM. This is a reordering model in the sense that this model uses the information embedded in the alignment and tells the decoder what would be an appropriate source word translation order. The experimental results show that our model is able to improve the translation quality. We also compare our method with other three models and the results illustrate that the source-side LM can give additional benefits based on the other three models.

For the future work, we plan to try several extensions.

- Use more bilingual data. As we mentioned in Subsection 5.1, the 9-gram LM is very sparse. We want to use more bilingual corpora to do the

source	美军 发言人 稍 早 表示 , 美军 所属 的 军医 院 传 报 了 eighteen 人 丧 生 , 28 人 受伤 ; 死者 中 有 sixteen 人 是 伊拉克 民众 , two 人 是 美国 国防部 的 美 籍 职员 .
base	earlier , a spokesman for the us military said that a us military hospital affiliated to the media has reported that eighteen people were killed and 28 injured ; sixteen were among the dead were iraqis and two american employees of the department of defense of the united states .
base+mero	a us military spokesman earlier said that us troops belonging to the military hospital , the newspaper reported that eighteen people were killed and 28 others were injured . among the dead there were sixteen iraqis , two american employees of the us department of defense .
base+sc	earlier , a spokesman for the us military said that a us military hospital affiliated to the reported eighteen people were killed and 28 others were injured . among the dead there were sixteen iraqis , two of the us department of defense is the american staff .
base+wolm	earlier , a spokesman for the us military said that a us military hospital , which is under the jurisdiction of the reported eighteen people were killed and 28 others were injured . among the dead there were sixteen iraqis and two american employees of the us department of defense .
base+sc+mero	a us military spokesman said earlier that a us military hospital affiliated to the media reported that eighteen people were killed and 28 others were injured . among the dead there were sixteen iraqis and two us defense department of american officials .
base+sc+mero+wolm	a us military spokesman earlier said that us troops belonging to the military hospital in response to a reported eighteen people were killed and 28 others were injured . among the dead there were sixteen iraqis and two us defense department of american officials .
reference	A US military spokesman said earlier that 18 deaths and 28 injuries were reported in US army hospitals. The dead included 16 Iraqi civilians and two US staff members from the US Defense Department.

Table 6: Translation examples

training. On the other hand, we will also use lower order source-side LM to see if it hurts the translation quality so that we know if the decoder can really benefit from those high order n-grams.

- Extend the LM to phrase level. Our source-side decoding sequence is counted on word level, but the decoder is a phrase-based SMT. We can extend the decoding sequence from word level to phrase level. However, the sparseness problem will be then more severe.

- Using other source information than words to train the LM. LM is in essence a Markov chain. Usually the LM is word-based as we use words as the states of the Markov chain. We can use other information as the states like part-of-speech tags or semantic tags. This is another way to deal with the sparseness problem as the size of the tag set is usually much smaller than the vocabulary size.

Acknowledgments

This work was partly realized as part of the Quaero Programme, funded by OSEO, French State agency for innovation, and also partly based upon work supported by the Defense Advanced Research Projects Agency (DARPA) under Contract No. HR0011-08-

C-0110. Any opinions, findings and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the DARPA.

The authors would like to give special thanks to anonymous reviewers for their valuable comments and suggestions.

References

Pi-Chuan Chang, Huihsin Tseng, Dan Jurafsky, and Christopher D. Manning. 2009. Discriminative reordering with chinese grammatical relations features. In *Proceedings of the NAACL-HLT-09 Workshop on Syntax and Structure in Statistical Translation*, pages 51–59, Morristown, NJ, USA, June.

Colin Cherry. 2008. Cohesive phrase-based decoding for statistical machine translation. In *Proceedings of ACL-08: HLT*, pages 72–80, Columbus, Ohio, June.

Marta R. Costa-jussà and José A. R. Fonollosa. 2008. Computing multiple weighted reordering hypotheses for a statistical machine translation phrase-based system. In *Proceedings of AMTA-08*, pages 1–7, Honolulu, Hawaii, October.

Reinhard Kneser and Hermann Ney. 1995. Improved backing-off for m-gram language modeling. In *Proceedings of ICASSP-95*, pages 49–52, Detroit, USA, May.

Kevin Knight. 1999. Decoding complexity in word-replacement translation models. *Comput. Linguist.*, 25(4):607–615.

Roger Levy and Christopher D. Manning. 2003. Is it harder to parse chinese, or the chinese treebank? In *Proceedings of ACL-03*, pages 439–446, Sapporo, Japan, July.

Sonja Nießen, Franz Josef Och, Gregor Leusch, and Hermann Ney. 2000. An evaluation tool for machine translation: Fast evaluation for MT research. In *LREC00*, pages 39–45, Athens, Greece, May.

Franz Josef Och and Hermann Ney. 2002. Discriminative training and maximum entropy models for statistical machine translation. In *Proceedings of ACL-02*, pages 295–302, Philadelphia, Pennsylvania, USA, July.

Franz Josef Och, Daniel Gildea, Sanjeev Khudanpur, Anoop Sarkar, Kenji Yamada, Alex Fraser, Shankar Kumar, Libin Shen, David Smith, Katherine Eng, Viren Jain, Zhen Jin, and Dragomir Radev. 2004. A smorgasbord of features for statistical machine translation. In *Proceedings of NAACL-HLT-04*, pages 161–168, Boston, Massachusetts, USA, May.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2001. Bleu: a method for automatic evaluation of machine translation. IBM Research Report RC22176 (W0109-022), IBM Research Division, Thomas J. Watson Research Center, P.O. Box 218, Yorktown Heights, NY 10598, September.

Matthew Snover, Bonnie Dorr, Richard Schwartz, John Makhoul, Linnea Micciulla, and Ralph Weischedel. 2005. A study of translation error rate with targeted human annotation. Technical Report LAMP-TR-126, CS-TR-4755, UMIACS-TR-2005-58, University of Maryland, College Park, MD.

Andreas Stolcke. 2002. Srilm - an extensible language modeling toolkit. In *Proceedings of ICSLP-02*, pages 901–904, Denver, Colorado,USA, September.

Christoph Tillmann, Stephan Vogel, Hermann Ney, Alex Zubiaga, and Hassan Sawaf. 1997. Accelerated DP based search for statistical translation. In *Proceedings of EUROSPEECH-97*, pages 2667–2670, Rhodes, Greece, September.

Chao Wang, Michael Collins, and Philipp Koehn. 2007. Chinese syntactic reordering for statistical machine translation. In *Proceedings of the EMNLP/CoNLL-07*, pages 737–745, Prague, Czech Republic, June.

Richard Zens and Hermann Ney. 2006. Discriminative reordering models for statistical machine translation. In *Proceedings of the Workshop on Statistical Machine Translation at HLT-NAACL-06*, pages 55–63, New York City, NY, June.

Richard Zens. 2008. *Phrase-based Statistical Machine Translation: Models, Search, Training.* Ph.D. thesis, RWTH, Aachen, February.

Yuqi Zhang, Richard Zens, and Hermann Ney. 2007. Chunk-level reordering of source language sentences with automatically learned rules for statistical machine translation. In *Proceedings of the NAACL-HLT-07/AMTA Workshop on Syntax and Structure in Statistical Translation*, pages 1–8, Morristown, NJ, USA, April.