Cut the noise: Mutually reinforcing reordering and alignments for improved machine translation

Karthik Visweswariah
IBM Research India
v-karthik@in.ibm.com

Mitesh M. Khapra
IBM Research India
mikhapra@in.ibm.com

Ananthakrishnan Ramanathan
IBM Research India
anandr42@gmail.com

Abstract

Preordering of a source language sentence to match target word order has proved to be useful for improving machine translation systems. Previous work has shown that a reordering model can be learned from high quality manual word alignments to improve machine translation performance. In this paper, we focus on further improving the performance of the reordering model (and thereby machine translation) by using a larger corpus of sentence aligned data for which manual word alignments are not available but automatic machine generated alignments are available. The main challenge we tackle is to generate quality data for training the reordering model in spite of the machine alignments being noisy. To mitigate the effect of noisy machine alignments, we propose a novel approach that improves reorderings produced given noisy alignments and also improves word alignments using information from the reordering model. This approach generates alignments that are 2.6 f-Measure points better than a baseline supervised aligner. The data generated allows us to train a reordering model that gives an improvement of 1.8 BLEU points on the NIST MT-08 Urdu-English evaluation set over a reordering model that only uses manual word alignments, and a gain of 5.2 BLEU points over a standard phrase-based baseline.

1 Introduction

Dealing with word order differences between source and target languages presents a significant challenge for machine translation systems. Failing to produce target words in the correct order results in machine translation output that is not fluent and is often very hard to understand. These problems are particularly severe when translating between languages which have very different structure.

Phrase based systems (Koehn et al., 2003) use lexicalized distortion models (Al-Onaizan and Papineni, 2006; Tillman, 2004) and scores from the target language model to produce words in the correct order in the target language. These systems typically are only able to capture short range reorderings and the amount of data required to potentially capture longer range reordering phenomena is prohibitively large.

There has been a large body of work showing the efficacy of preordering source sentences using a source parser and applying hand written or automatically learned rules (Collins et al., 2005; Wang et al., 2007; Ramanathan et al., 2009; Xia and McCord, 2004; Genzel, 2010; Visweswariah et al., 2010). Recently, approaches that address the problem of word order differences between the source and target language without requiring a high quality source or target parser have been proposed (DeNero and Uszkoreit, 2011; Visweswariah et al., 2011; Neubig et al., 2012). These methods use a small corpus of manual word alignments (where the words in the source sentence are manually aligned to the words in the target sentence) to learn a model to preorder the source sentence to match target order.

In this paper, we build upon the approach in (Visweswariah et al., 2011) which uses manual word alignments for learning a reordering model. Specifically, we show that we can significantly improve reordering performance by using a large number of sentence pairs for which manual word alignments are not available. The motivation for going beyond manual word alignments is clear: the reordering model can have millions of features and estimating weights for the features on thousands of sentences of manual word alignments is

Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics, pages 1275–1284,
Sofia, Bulgaria, August 4-9 2013.

likely to be inadequate. One approach to deal with this problem would be to use only part-of-speech tags as features for all but the most frequent words. This will cut down on the number of features and perhaps the model would be learnable with a small set of manual word alignments. Unfortunately, as we will see in the experimental section, leaving out lexical information from the models hurts performance even with a relatively small set of manual word alignments. Another option would be to collect more manual word alignments but this is undesirable because it is time consuming and expensive.

The challenge in going beyond manual word alignments and using machine alignments is the noise in the machine alignments which affects the performance of the reordering model (see Section 5). We illustrate this with the help of a motivating example. Consider the example English sentence and its translation shown in Figure 1.

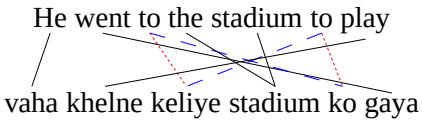

Figure 1: An example English sentence with its Urdu translation with alignment links. Red (dotted) links are incorrect links while the blue (dashed) links are the corresponding correct links.

A standard word alignment algorithm that we used (McCarley et al., 2011) made the mistake of mis-aligning the Urdu *ko* and *keliye* (it switched the two). Deriving reference reorderings from these wrong alignments would give us an incorrect reordering. A reordering model trained on such incorrect reorderings would obviously perform poorly. Our task is thus two-fold (i) improve the quality of machine alignments (ii) use these less noisy alignments to derive cleaner training data for a reordering model.

Before proceeding, we first point out that the two tasks, *viz.*, reordering and word alignment are related: Having perfect reordering makes the alignment task easier while having perfect alignments in turn makes the task of finding reorderings trivial. Motivated by this fact, we introduce models that allow us to connect the source/target reordering and the word alignments and show that these models help in mutually improving the performance of word alignments and reordering. Specifically, we build two models: the first scores reorderings given the source sentence and noisy alignments, the second scores alignments given the noisy source and target reorderings and the source and target sentences themselves. The second model helps produce better alignments, while we use the first model to help generate better reference reordering given noisy alignments. These improved reference reorderings will then be used to train a reordering model.

Our experiments show that reordering models trained using these improved machine alignments perform significantly better than models trained only on manual word alignments. This results in a 1.8 BLEU point gain in machine translation performance on an Urdu-English machine translation task over a preordering model trained using only manual word alignments. In all, this increases the gain in performance by using the preordering model to 5.2 BLEU points over a standard phrase-based system with no preordering.

The rest of this paper is structured as follows. Section 2 describes the main reordering issues in Urdu-English translation. Section 3 introduces the reordering modeling framework that forms the basis for our work. Section 4 describes the two models we use to tie together reordering and alignments and how we use these models to generate training data for training our reordering model. Section 5 presents the experimental setup used for evaluating the models proposed in this paper on an Urdu-English machine translation task. Section 6 presents the results of our experiments. We describe related work in Section 7 and finally present some concluding remarks and potential future work in Section 8.

2 Reordering issues in Urdu-English translation

In this section we describe the main sources of word order differences between Urdu and English since this is the language pair we experiment with in this paper.

The typical word order in Urdu is *Subject-Object-Verb* unlike English in which the order is *Subject-Verb-Object*. Urdu has case markers that sometimes (but not always) mark the subject and the object of a sentence. This difference in the placement of verbs can often lead to movements of verbs over long distances (depending on the number of words in the object). Phrase based systems do not capture such long distance movements well.

Another difference is that Urdu uses post-positions unlike English which uses prepositions. This can also lead to long range movements depending on the length of the noun phrase that the post-position follows. The order of noun phrases and prepositional phrases is also swapped in Urdu as compared with English.

3 Reordering model

In this section we briefly describe the reordering model (Visweswariah et al., 2011) that forms the basis of our work. We also describe an approximation we make in the training process that significantly speeds up the training without much loss of accuracy which enables training on much larger data sets. Consider a source sentence $\mathbf{w}$ that we would like to reorder to match the target order. Let $\boldsymbol{\pi}$ represent a candidate permutation of the source sentence $\mathbf{w}$. π_i denotes the index of the word in the source sentence that maps to position i in the candidate reordering, thus reordering with this candidate permutation $\boldsymbol{\pi}$ we will reorder the sentence $\mathbf{w}$ to $w_{\pi_1}, w_{\pi_2}, ..., w_{\pi_n}$. The reordering model we use assigns costs to candidate permutations as:

$$C(\boldsymbol{\pi}|\mathbf{w}) = \sum_i c(\pi_{i-1}, \pi_i).$$

The costs $c(m, n)$ are pairwise costs of putting w_m immediately before w_n in the reordering. We reorder the sentence $\mathbf{w}$ according to the permutation $\boldsymbol{\pi}$ that minimizes the cost $C(\boldsymbol{\pi}|\mathbf{w})$. We find the minimal cost permutation by converting the problem into a symmetric Travelling Salesman Problem (TSP) and then using an implementation of the chained Lin-Kernighan heuristic (Applegate et al., 2003). The costs in the reordering model $c(m, n)$ are parameterized by a linear model:

$$c(m, n) = \boldsymbol{\theta}^T \Phi(\mathbf{w}, m, n)$$

where $\boldsymbol{\theta}$ is a learned vector of weights and Φ is a vector of binary feature functions that inspect the words and POS tags of the source sentence at and around positions m and n. We use the features (Φ) described in Visweswariah et al. (2011) that were based on features used in dependency parsing (McDonald et al., 2005a).

To learn the weight vector $\boldsymbol{\theta}$ we require a corpus of sentences $\mathbf{w}$ with their desired reorderings $\boldsymbol{\pi}^*$. Past work Visweswariah et al. (2011) used high quality manual word alignments to derive the desired reorderings $\boldsymbol{\pi}^*$ as follows. Given word aligned source and target sentences, we drop the source words that are not aligned[1]. Let m_i be the mean of the target word positions that the source word at index i is aligned to. We then sort the source indices in increasing order of m_i (this order defines $\boldsymbol{\pi}^*$). If $m_i = m_j$ (for example, because w_i and w_j are aligned to the same set of words) we keep them in the same order that they occurred in the source sentence.

We used the single best Margin Infused Relaxed Algorithm (MIRA) (McDonald et al. (2005b), Crammer and Singer (2003)) with online updates to our parameters given by:

$$\begin{aligned} \boldsymbol{\theta}_{i+1} = & \quad \arg\min_{\boldsymbol{\theta}} ||\boldsymbol{\theta} - \boldsymbol{\theta}_i|| \\ & s.t. \quad C(\boldsymbol{\pi}^*|\mathbf{w}) < C(\hat{\boldsymbol{\pi}}|\mathbf{w}) - L(\boldsymbol{\pi}^*, \hat{\boldsymbol{\pi}}). \end{aligned}$$

In the equation above, $\hat{\boldsymbol{\pi}} = \arg\min_{\boldsymbol{\pi}} C(\boldsymbol{\pi}|\mathbf{w})$ is the best reordering based on the current parameter value $\boldsymbol{\theta}_i$ and L is a loss function. We take L to be the number of words for which the hypothesized permutation $\hat{\boldsymbol{\pi}}$ has a different preceding word as compared with the reference permutation $\boldsymbol{\pi}^*$.

In this paper we focus on the case where in addition to using a relatively small number of manual word aligned sentences to derive the reference permutations $\boldsymbol{\pi}^*$ used to train our model, we would like to use more abundant but noisier machine aligned sentence pairs. To handle the larger amount of training data we obtain from machine alignments, we make an approximation in training that we found empirically to not affect performance but that makes training faster by more than a factor of five. This allows us to train the reordering model with roughly 150K sentences in about two hours. The approximation we make is that instead of using the chained Lin-Kernighan heuristic to solve the TSP problem to find $\hat{\boldsymbol{\pi}} = \arg\min_{\boldsymbol{\pi}} C(\boldsymbol{\pi}|\mathbf{w})$, we select greedily for each word the preceding word that has the lowest cost[2]. Using ψ_i to denote $\arg\min_j c(j, i)$ and letting

$$C(\boldsymbol{\psi}|\mathbf{w}) = \sum_i c(\psi_i, i),$$

[1] Note that the unaligned source words are dropped only at the time of training. At the time of testing all source words are retained as the alignment information is obviously not available at test time.

[2] It should be noted that this approximation was done only at the time of training. At the time of testing we still use the chained Lin-Kernighan heuristic to solve the TSP problem.

we do the update according to:

$$\boldsymbol{\theta}_{i+1} = \arg\min_{\boldsymbol{\theta}} ||\boldsymbol{\theta} - \boldsymbol{\theta}_i|| \quad s.t. \quad C(\boldsymbol{\pi}^*|\mathbf{w}) < C(\boldsymbol{\psi}|\mathbf{w}) - L(\boldsymbol{\pi}^*, \boldsymbol{\psi}).$$

Again the loss $L(\boldsymbol{\pi}^*, \boldsymbol{\psi})$ is the number of positions i for which π^*_{i-1} is different from ψ_{i-1}.

4 Generating reference reordering from parallel sentences

The main aim of our work is to improve the reordering model by using parallel sentences for which manual word alignments are not available. In other words, we want to generate relatively clean reference reorderings from parallel sentences and use them for training a reordering model. A straightforward approach for this is to use a supervised aligner to align the words in the sentences and then derive the reference reordering as we do for manual word alignments. However, as we will see in the experimental results, the quality of a reordering model trained from automatic alignments is very sensitive to the quality of alignments. This motivated us to explore if we can further improve our aligner and the method for generating reference reorderings given alignments.

We improve upon the above mentioned basic approach by coupling the tasks of reordering and word alignment. We do this by building a **reordering model** ($C(\boldsymbol{\pi^s}|\mathbf{w^s}, \mathbf{w^t}, \mathbf{a})$) that scores reorderings $\boldsymbol{\pi^s}$ given the source sentence $\mathbf{w^s}$, target sentence $\mathbf{w^t}$ and machine alignments $\mathbf{a}$. Complementing this model, we build an **alignment model** ($P(\mathbf{a}|\mathbf{w^s}, \mathbf{w^t}, \boldsymbol{\pi^s}, \boldsymbol{\pi^t})$) that scores alignments $\mathbf{a}$ given the source and target sentences and their predicted reorderings according to source and target reordering models. The model ($C(\boldsymbol{\pi^s}|\mathbf{w^s}, \mathbf{w^t}, \mathbf{a})$) helps to produce better reference reorderings for training our final reordering model given fixed machine alignments and the alignment model ($P(\mathbf{a}|\mathbf{w^s}, \mathbf{w^t}, \boldsymbol{\pi^s}, \boldsymbol{\pi^t})$) helps improve the machine alignments taking into account information from reordering models. In the following sections, we describe our overall approach followed by a description of the two models.

4.1 Overall approach to generating training data

We first describe our overall approach to generating training data for the reordering model given a small corpus of sentences with manual word alignments (H) and a much larger corpus of parallel sentences (U) that are not word aligned. The basic idea is to chain together the two models, *viz.*, reordering model and alignment model, as illustrated in Figure 2. The steps involved are as described below:

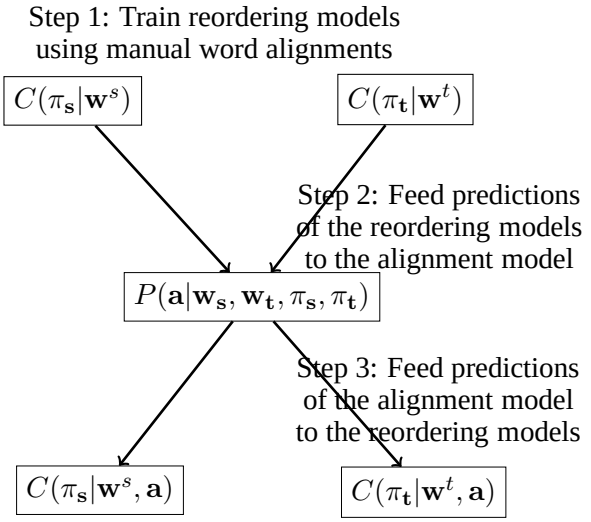

Figure 2: Overall approach: Building a sequence of reordering and alignment models.

Step 1: First, we use manual word alignments (H) to train source and target reordering models as described in (Visweswariah et al., 2011).

Step 2: Next, we use the hand alignments to train an alignment model $P(\mathbf{a}|\mathbf{w^s}, \mathbf{w^t}, \boldsymbol{\pi^s}, \boldsymbol{\pi^t})$. In addition to the original source and target sentence, we also feed the predictions of the reordering model trained in Step 1 to this alignment model (see section 4.2 for details of the model itself).

Step 3: Finally, we use the predictions of the alignment model trained in Step 2 to train reordering models $C(\boldsymbol{\pi^s}|\mathbf{w^s}, \mathbf{w^t}, \mathbf{a})$ (see section 4.3 for details on the reordering model itself).

After building the sequence of models shown in Figure 2, we apply them in sequence on the unaligned parallel data U, starting with the reordering models $C(\boldsymbol{\pi^s}|\mathbf{w^s})$ and $C(\boldsymbol{\pi^t}|\mathbf{w^t})$. The reorderings obtained for the source side in U (after applying the final model $C(\boldsymbol{\pi^s}|\mathbf{w^s}, \mathbf{a})$) are used along with reference reorderings obtained from the manual word alignments to train our reordering model. Note that, in theory, we could iterate over steps 2 and 3 several times but, in practice we did not see a benefit of going beyond one iter-

ation in our experiments. Also, since we are interested only in the source side reorderings produced by the model $C(\boldsymbol{\pi^s}|\mathbf{w^s}, \mathbf{a})$, the target reordering model $C(\boldsymbol{\pi^t}|\mathbf{w^t}, \mathbf{a})$ is needed only if we iterate over steps 2 and 3.

We now point to some practical considerations of our approach. Consider the case when we are training an alignment model conditioned on reorderings ($P(\mathbf{a}|\mathbf{w^s}, \mathbf{w^t}, \boldsymbol{\pi^s}, \boldsymbol{\pi^t})$). If the reordering model that generated these reorderings $\boldsymbol{\pi^s}, \boldsymbol{\pi^t}$ were trained on the same data that we are using to train the alignment model, then the reorderings would be much better than we would expect on unseen test data, and hence the alignment model ($P(\mathbf{a}|\mathbf{w^s}, \mathbf{w^t}, \boldsymbol{\pi^s}, \boldsymbol{\pi^t})$) may learn to make the alignment overly consistent with the reorderings $\boldsymbol{\pi^s}$ and $\boldsymbol{\pi^t}$. To counter this problem, we divide the training data H into K parts and at each stage we apply a model (reordering or alignment) on part i that had not seen part i in training. This ensures that the alignment model does not see very optimistic reorderings and vice versa. We now describe the individual models, *viz.*, $P(\mathbf{a}|\mathbf{w^s}, \mathbf{w^t}, \boldsymbol{\pi^s}, \boldsymbol{\pi^t})$ and $C(\boldsymbol{\pi^s}|\mathbf{w^s}, \mathbf{a})$.

4.2 Modeling alignments given reordering

In this section we describe how we fuse information from source and target reordering models to improve word alignments.

As a base model we use the correction model for word alignments proposed by McCarley et al. (2011). This model was significantly better than the MaxEnt aligner (Ittycheriah and Roukos, 2005) and is also flexible in the sense that it allows for arbitrary features to be introduced while still keeping training and decoding tractable by using a greedy decoding algorithm that explores potential alignments in a small neighborhood of the current alignment. The model thus needs a reasonably good initial alignment to start with for which we use the MaxEnt aligner (Ittycheriah and Roukos, 2005) as in McCarley et al. (2011).

The correction model is a log-linear model:

$$P(\mathbf{a}|\mathbf{w^s}, \mathbf{w^t}) = \frac{\exp(\lambda^T \boldsymbol{\phi}(\mathbf{a}, \mathbf{w^s}, \mathbf{w^t}))}{Z(\mathbf{w^s}, \mathbf{w^t})}.$$

The λs are trained using the LBFGS algorithm (Liu et al., 1989) to maximize the log-likelihood smoothed with L_2 regularization. The feature functions $\boldsymbol{\phi}$ we start with are those used in McCarley et al. (2011) and include features encoding the Model 1 probabilities between pairs of words linked in the alignment $\mathbf{a}$, features that inspect source and target POS tags and parses (if available) and features that inspect the alignments of adjacent words in the source and target sentence.

To incorporate information from the reordering model, we add features that use the predicted source $\boldsymbol{\pi^s}$ and target permutations $\boldsymbol{\pi^t}$. We introduce some notation to describe these features. Let S_m and S_n be the set of indices of target words that w^s_m and w^s_n are aligned to respectively. We define the minimum signed distance (msd) between these two sets as:

$$msd(S_m, S_n) = i^* - j^*$$

where, $$(i^*, j^*) = \arg \min_{(i,j) \in S_m \times S_n} |i - j|$$

We quantize and encode with binary features the minimum signed distance between the sets of the indices of the target words that source words adjacent in the reordering $\boldsymbol{\pi}^s$ ($w^s_{\pi^s_i}$ and $w^s_{\pi^s_{i+1}}$) are aligned to. We instantiate similar features with the roles of source and target sentences reversed. With this addition of features we use the same training and testing procedure as in McCarley et al. (2011). If the reorderings $\boldsymbol{\pi}^s$ were perfect we would learn to only allow alignments where $w^s_{\pi^s_i}$ and $w^s_{\pi^s_{i+1}}$ were aligned to adjacent words in the target sentence. Although the reordering model is not perfect, preferring alignments consistent with the reordering models improves the aligner.

4.3 Modeling reordering given alignments

To model source permutations given source ($\mathbf{w^s}$) and target ($\mathbf{w^t}$) sentences, and alignments ($\mathbf{a}$) we reuse the reordering model framework described in Section 3 adding additional features capturing the relation between a hypothesized permutation $\boldsymbol{\pi}$ and alignments $\mathbf{a}$. To allow for searching via the same TSP formulation we once again assign costs to candidate permutations as:

$$C(\boldsymbol{\pi^s}|\mathbf{w^s}, \mathbf{w^t}, \mathbf{a}) = \sum_i c(\pi_{i-1}, \pi_i|\mathbf{w^s}, \mathbf{a}).$$

Note that we introduce a dependence on the target sentence $\mathbf{w^t}$ only through the alignment $\mathbf{a}$. Once again we parameterize the costs by a linear model:

$$c(m, n) = \boldsymbol{\theta}^T \boldsymbol{\Phi}(\mathbf{w^s}, \mathbf{a}, m, n).$$

For the feature functions $\boldsymbol{\Phi}$, in addition to the features that only depend on $\mathbf{w^s}, m, n$ (that we

use in our standard reordering model) we add binary indicator features based on $msd(S_m, S_n)$ and $msd(S_m, S_n)$ conjoined with POS(w_m^s) and POS(w_n^s).

Here, S_m and S_n are the set of indices of target words that w_m^s and w_n^s are aligned to respectively. We conjoin the msd (minimum signed distance) with the POS tags to allow the model to capture the fact that the alignment error rate maybe higher for some POS tags than others (*e.g.*, we have observed verbs have a higher error rate in Urdu-English alignments).

Given these features we train the parameters $\boldsymbol{\theta}$ using the MIRA algorithm as described in Section 3. Using this model, we can find the lowest cost permutation $C(\boldsymbol{\pi^s}|\mathbf{w^s}, \mathbf{a})$ using the Lin-Kernighan heuristic as described in Section 3. This model allows us to combine features from the original reordering model along with information coming from the alignments to find source reorderings given a parallel corpus and alignments. We will see in the experimental section that this improves upon the simple heuristic for deriving reorderings described in Section 3.

5 Experimental setup

In this section we describe the experimental setup that we used to evaluate the models proposed in this paper. All experiments were done on Urdu-English and we evaluate reordering in two ways: Firstly, we evaluate reordering performance directly by comparing the reordered source sentence in Urdu with a reference reordering obtained from the manual word alignments using BLEU (Papineni et al., 2002) (we call this measure monolingual BLEU or mBLEU). All mBLEU results are reported on a small test set of about 400 sentences set aside from our set of sentences with manual word alignments. Additionally, we evaluate the effect of reordering on our final systems for machine translation measured using BLEU.

We use about 10K sentences (180K words) of manual word alignments which were created in house using part of the NIST MT-08 training data[3] to train our baseline reordering model and to train our supervised machine aligners. We use a parallel corpus of 3.9M words consisting of 1.7M words from the NIST MT-08 training data set and 2.2M words extracted from parallel news stories on the web[4]. The parallel corpus is used for building our phrased based machine translation system and to add training data for our reordering model. For our English language model, we use the Gigaword English corpus in addition to the English side of our parallel corpus. Our Part-of-Speech tagger is a Maximum Entropy Markov model tagger trained on roughly fifty thousand words from the CRULP corpus (Hussain, 2008).

For our machine translation experiments, we used a standard phrase based system (Al-Onaizan and Papineni, 2006) with a lexicalized distortion model with a window size of +/-4 words[5]. To extract phrases we use HMM alignments along with higher quality alignments from a supervised aligner (McCarley et al., 2011). We report results on the (four reference) NIST MT-08 evaluation set in Table 4 for the News and Web conditions. The News and Web conditions each contain roughly 20K words in the test set, with the Web condition containing more informal text from the web.

6 Results and Discussions

We now discuss the results of our experiments.

Need for additional data: We first show the need for additional data in Urdu-English reordering. Column 2 of Table 1 shows mBLEU as a function of the number of sentences with manual word alignments that are used to train the reordering model. We see a roughly 3 mBLEU points drop in performance per halving of data indicating a potential for improvement by adding more data.

Using fewer features: We compare the performance of a model trained using lexical features for all words (Column 2 of Table 1) with a model trained using lexical features only for the 1000 most frequent words (Column 3 of Table 1). The motivation for this is to explore if a good model can be learned even from a small amount of data if we restrict the number of features in a reasonable manner. However, we see that even with only 2.4K sentences with manual word alignments our model benefits from lexical identities of more than the 1000 most frequent words.

Effect of quality of machine alignments: We next look at the use of automatically generated

[3]http://www.ldc.upenn.edu

[4]http://centralasiaonline.com

[5]Note that the same window size of +/-4 words was used for all the systems, *i.e.*, the baseline system as well as the systems using different preordering techniques.

Data size	All features	Frequent lex only
10K	52.5	50.8
5K	49.6	49.0
2.5K	46.6	46.2

Table 1: mBLEU scores for Urdu to English reordering using different number of sentences of manually word aligned training data with all features and with lexical features instantiated only for the 1000 most frequent words.

machine alignments to train the reordering model and see the effect of aligner quality on the reordering model generated using this data. These experiments also form the baseline for the models we propose in this paper to clean up alignments. We experimented with two different supervised aligners : a maximum entropy aligner (Ittycheriah and Roukos, 2005) and an improved correction model that corrects the maximum entropy alignments (McCarley et al., 2011).

Aligner		Train size (words)	mBLEU
Type	f-Measure		
None		-	35.5
Manual		180K	52.5
MaxEnt	70.0	3.9M	49.5
Correction model	78.1	3.9M	55.1

Table 2: mBLEU scores for Urdu to English reordering using models trained on different data sources and tested on a development set of 8017 Urdu tokens.

Table 2 shows mBLEU scores when the reordering model is trained on reordering references created from aligners with different quality. We see that the quality of the alignments matter a great deal to the reordering model; using MaxEnt alignments cause a degradation in performance over just using a small set of manual word alignments. The alignments obtained using the aligner of McCarley et al. (2011) are of much better quality and hence give higher reordering performance. Note that this reordering performance is much better than that obtained using manual word alignments because the size of machine alignments is much larger (3.9M v/s 180K words).

Improvements in reordering performance using the proposed models: Table 3 shows improvements in the reordering model when using the models proposed in this paper. We use H to refer to the manually word aligned data and U to refer to the additional sentence pairs for which manual word alignments are not available. We report the following numbers :

1. Base correction model: This is the baseline where we use the correction model of McCarley et al. (2011) for generating word alignments. The f-Measure of this aligner is 78.1% (see row 1, column 2). Corresponding to this, we also report the baseline for our reordering experiments in the third column. Here, we first generate word alignments for U using the aligner of McCarley et al. (2011) and then extract reference reorderings from these alignments. We then combine these reference reorderings with the reference reorderings derived from H and use this combined data to train a reordering model which serves as the baseline (mBLEU = 55.1).

2. Correction model, $C(\boldsymbol{\pi}|\mathbf{a})$: Here, once again we generate alignments for U using the correction model of McCarley et al. (2011). However, instead of using the basic approach of extracting reference reorderings, we use our improved model $C(\boldsymbol{\pi}|\mathbf{a})$ to generate reference reorderings from U. These reference reorderings are again combined with the reference reorderings derived from H and used to train a reordering model (mBLEU = 56.4).

3. $P(\mathbf{a}|\boldsymbol{\pi})$**,** $C(\boldsymbol{\pi}|\mathbf{a})$: Here, we build the entire sequence of models shown in Figure 2. The alignment model $P(\mathbf{a}|\boldsymbol{\pi})$ is first improved by using predictions from the reordering model. These improved alignments are then used to extract better reference reorderings from U using $C(\boldsymbol{\pi}|\mathbf{a})$.

We see substantial improvements over simply adding in the data from the machine alignments. Improvements come roughly in equal parts from the two techniques we proposed in this paper : (i) using a model to generate reference reorderings from noisy alignments and (ii) using reordering information to improve the aligner.

Method	f-Measure	mBLEU
Base Correction model	78.1	55.1
Correction model, $C(\boldsymbol{\pi}\|\mathbf{a})$	78.1	56.4
$P(\mathbf{a}\|\boldsymbol{\pi})$, $C(\boldsymbol{\pi}\|\mathbf{a})$	80.7	57.6

Table 3: mBLEU with different methods to generate reordering model training data from a machine aligned parallel corpus in addition to manual word alignments.

Improvements in MT performance using the proposed models: We report results for a phrase based system with different preordering techniques. For results including a reordering model, we simply reorder the source side Urdu data both while training and at test time. In addition to

phrase based systems with different preordering methods, we also report on a hierarchical phrase based system for which we used Joshua 4.0 (Ganitkevitch et al., 2012). We see a significant gain of 1.8 BLEU points in machine translation by going beyond manual word alignments using the best reordering model reported in Table 3. We also note a gain of 2.0 BLEU points over a hierarchical phrase based system.

System type	MT-08 eval		
	Web	News	All
Baseline (no preordering)	18.4	25.6	22.2
Hierarchical phrase based	19.6	30.7	25.4
Reordering: Manual alignments	20.7	30.0	25.6
+ Machine alignments simple	21.3	30.9	26.4
+ machine alignments, model based	**22.1**	**32.2**	**27.4**

Table 4: MT performance without preordering (phrase based and hierarchical phrase based), and with reordering models using different data sources (phrase based).

7 Related work

Dealing with the problem of handling word order differences in machine translation has recently received much attention. The approaches proposed for solving this problem can be broadly divided into 3 sets as discussed below.

The first set of approaches handle the reordering problem as part of the decoding process. Hierarchical models (Chiang, 2007) and syntax based models (Yamada and Knight, 2002; Galley et al., 2006; Liu et al., 2006; Zollmann and Venugopal, 2006) improve upon the simpler phrase based models but with significant additional computational cost (compared with phrase based systems) due to the inclusion of chart based parsing in the decoding process. Syntax based models also require a high quality source or target language parser.

The second set of approaches rely on a source language parser and treat reordering as a separate process that is applied on the source language sentence at training and test time before using a standard approach to machine translation. Preordering the source data with hand written or automatically learned rules is effective and efficient (Collins et al., 2005; Wang et al., 2007; Ramanathan et al., 2009; Xia and McCord, 2004; Genzel, 2010; Visweswariah et al., 2010) but requires a source language parser.

Recent approaches that avoid the need for a source or target language parser and retain the efficiency of preordering models were proposed in (Tromble and Eisner, 2009; DeNero and Uszkoreit, 2011; Visweswariah et al., 2011; Neubig et al., 2012). (DeNero and Uszkoreit, 2011; Visweswariah et al., 2011; Neubig et al., 2012) focus on the use of manual word alignments to learn preordering models and in both cases no benefit was obtained by using the parallel corpus in addition to manual word alignments. Our work is an extension of Visweswariah et al. (2011) and we focus on being able to incorporate relatively noisy machine alignments to improve the reordering model.

In addition to being related to work in reordering, our work is also more broadly related to several other efforts which we now outline. Setiawan et al. (2010) proposed the use of function word reordering to improve alignments. While this work is similar to one of our models (model of alignments given reordering) we differ in using a reordering model of all words (not just function words) and both source and target sentences (not just the source sentence). The task of directly learning a reordering model for language pairs that are very different is closely related to the task of parsing and hence work on semi-supervised parsing (Koo et al., 2008; McClosky et al., 2006; Suzuki et al., 2009) is broadly related to our work. Our work coupling reordering and alignments is also similar in spirit to approaches where parsing and alignment are coupled (Wu, 1997).

8 Conclusion

In the paper we showed that a reordering model can benefit from data beyond a relatively small corpus of manual word alignments. We proposed a model that scores reorderings given alignments and the source sentence that we use to generate cleaner training data from noisy alignments. We also proposed a model that scores alignments given source and target sentence reorderings that improves a supervised alignment model by 2.6 points in f-Measure. While the improvement in alignment performance is modest, the improvement does result in improved reordering models. Cumulatively, we see a gain of 1.8 BLEU points over a baseline reordering model that only uses manual word alignments, a gain of 2.0 BLEU points over a hierarchical phrase based system, and a gain of 5.2 BLEU points over a phrase based

system that uses no source preordering on a publicly available Urdu-English test set.

As future work we would like to evaluate our models on other language pairs. Another avenue of future work we would like to explore is the use of monolingual source and target data to further assist the reordering model. We hope to be able to learn lexical information such as how many arguments a verb takes, what nouns are potential subjects for a given verb by gathering statistics from an English parser and projecting to the source language via our word/phrase translation table.

References

Yaser Al-Onaizan and Kishore Papineni. 2006. Distortion models for statistical machine translation. In *Proceedings of ACL*, ACL-44, pages 529–536, Morristown, NJ, USA. Association for Computational Linguistics.

David Applegate, William Cook, and Andre Rohe. 2003. Chained lin-kernighan for large traveling salesman problems. In *INFORMS Journal On Computing*.

David Chiang. 2007. Hierarchical phrase-based translation. *Comput. Linguist.*, 33(2):201–228, June.

Michael Collins, Philipp Koehn, and Ivona Kučerová. 2005. Clause restructuring for statistical machine translation. In *Proceedings of ACL*, pages 531–540, Morristown, NJ, USA. Association for Computational Linguistics.

Koby Crammer and Yoram Singer. 2003. Ultraconservative online algorithms for multiclass problems. *J. Mach. Learn. Res.*, 3:951–991, March.

John DeNero and Jakob Uszkoreit. 2011. Inducing sentence structure from parallel corpora for reordering. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, EMNLP '11, pages 193–203, Stroudsburg, PA, USA. Association for Computational Linguistics.

Michel Galley, Jonathan Graehl, Kevin Knight, Daniel Marcu, Steve DeNeefe, Wei Wang, and Ignacio Thayer. 2006. Scalable inference and training of context-rich syntactic translation models. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics*, ACL-44, pages 961–968, Stroudsburg, PA, USA. Association for Computational Linguistics.

Juri Ganitkevitch, Yuan Cao, Jonathan Weese, Matt Post, and Chris Callison-Burch. 2012. Joshua 4.0: Packing, pro, and paraphrases. In *Proceedings of the Seventh Workshop on Statistical Machine Translation*, pages 283–291, Montréal, Canada, June. Association for Computational Linguistics.

Dmitriy Genzel. 2010. Automatically learning source-side reordering rules for large scale machine translation. In *Proceedings of the 23rd International Conference on Computational Linguistics*.

Sarmad Hussain. 2008. Resources for Urdu language processing. In *Proceedings of the 6th Workshop on Asian Language Resources, IJCNLP'08*.

Abraham Ittycheriah and Salim Roukos. 2005. A maximum entropy word aligner for Arabic-English machine translation. In *Proceedings of HLT/EMNLP*, HLT '05, pages 89–96, Stroudsburg, PA, USA. Association for Computational Linguistics.

Philipp Koehn, Franz Josef Och, and Daniel Marcu. 2003. Statistical phrase-based translation. In *Proceedings of HLT-NAACL*.

Terry Koo, Xavier Carreras, and Michael Collins. 2008. Simple semi-supervised dependency parsing. In *ACL*, pages 595–603.

Dong C. Liu, Jorge Nocedal, and Dong C. 1989. On the limited memory bfgs method for large scale optimization. *Mathematical Programming*, 45:503–528.

Yang Liu, Qun Liu, and Shouxun Lin. 2006. Tree-to-string alignment template for statistical machine translation. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics*, ACL-44, pages 609–616, Stroudsburg, PA, USA. Association for Computational Linguistics.

J. Scott McCarley, Abraham Ittycheriah, Salim Roukos, Bing Xiang, and Jian-ming Xu. 2011. A correction model for word alignments. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, EMNLP '11, pages 889–898, Stroudsburg, PA, USA. Association for Computational Linguistics.

David McClosky, Eugene Charniak, and Mark Johnson. 2006. Effective self-training for parsing. In *HLT-NAACL*.

Ryan McDonald, Koby Crammer, and Fernando Pereira. 2005a. Online large-margin training of dependency parsers. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, ACL '05, pages 91–98, Stroudsburg, PA, USA. Association for Computational Linguistics.

Ryan McDonald, Fernando Pereira, Kiril Ribarov, and Jan Hajič. 2005b. Non-projective dependency parsing using spanning tree algorithms. In *Proceedings of HLT*.

Graham Neubig, Taro Watanabe, and Shinsuke Mori. 2012. Inducing a discriminative parser to optimize machine translation reordering. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational*

Natural Language Learning, pages 843–853, Jeju Island, Korea, July. Association for Computational Linguistics.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, ACL '02, pages 311–318, Stroudsburg, PA, USA. Association for Computational Linguistics.

Ananthakrishnan Ramanathan, Hansraj Choudhary, Avishek Ghosh, and Pushpak Bhattacharyya. 2009. Case markers and morphology: addressing the crux of the fluency problem in English-Hindi smt. In *Proceedings of ACL-IJCNLP*.

Hendra Setiawan, Chris Dyer, and Philip Resnik. 2010. Discriminative word alignment with a function word reordering model. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, EMNLP '10, pages 534–544, Stroudsburg, PA, USA. Association for Computational Linguistics.

Jun Suzuki, Hideki Isozaki, Xavier Carreras, and Michael Collins. 2009. An empirical study of semi-supervised structured conditional models for dependency parsing. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 2 - Volume 2*, EMNLP '09, pages 551–560, Stroudsburg, PA, USA. Association for Computational Linguistics.

Christoph Tillman. 2004. A unigram orientation model for statistical machine translation. In *Proceedings of HLT-NAACL*.

Roy Tromble and Jason Eisner. 2009. Learning linear ordering problems for better translation. In *Proceedings of EMNLP*.

Karthik Visweswariah, Jiri Navratil, Jeffrey Sorensen, Vijil Chenthamarakshan, and Nandakishore Kambhatla. 2010. Syntax based reordering with automatically derived rules for improved statistical machine translation. In *Proceedings of the 23rd International Conference on Computational Linguistics*.

Karthik Visweswariah, Rajakrishnan Rajkumar, Ankur Gandhe, Ananthakrishnan Ramanathan, and Jiri Navratil. 2011. A word reordering model for improved machine translation. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, EMNLP '11, pages 486–496, Stroudsburg, PA, USA. Association for Computational Linguistics.

Chao Wang, Michael Collins, and Philipp Koehn. 2007. Chinese syntactic reordering for statistical machine translation. In *Proceedings of EMNLP-CoNLL*.

Dekai Wu. 1997. Stochastic inversion transduction grammars and bilingual parsing of parallel corpora. *Comput. Linguist.*, 23(3):377–403, September.

Fei Xia and Michael McCord. 2004. Improving a statistical MT system with automatically learned rewrite patterns. In *COLING*.

Kenji Yamada and Kevin Knight. 2002. A decoder for syntax-based statistical MT. In *Proceedings of ACL*.

Andreas Zollmann and Ashish Venugopal. 2006. Syntax augmented machine translation via chart parsing. In *Proceedings on the Workshop on Statistical Machine Translation*.