Dynamically Shaping the Reordering Search Space of Phrase-Based Statistical Machine Translation

Arianna Bisazza and **Marcello Federico**
Fondazione Bruno Kessler
Trento, Italy
{bisazza,federico}@fbk.eu

Abstract

Defining the reordering search space is a crucial issue in phrase-based SMT between distant languages. In fact, the optimal trade-off between accuracy and complexity of decoding is nowadays reached by harshly limiting the input permutation space. We propose a method to dynamically shape such space and, thus, capture long-range word movements without hurting translation quality nor decoding time. The space defined by loose reordering constraints is dynamically pruned through a binary classifier that predicts whether a given input word should be translated right after another. The integration of this model into a phrase-based decoder improves a strong Arabic-English baseline already including state-of-the-art early distortion cost (Moore and Quirk, 2007) and hierarchical phrase orientation models (Galley and Manning, 2008). Significant improvements in the reordering of verbs are achieved by a system that is notably faster than the baseline, while BLEU and METEOR remain stable, or even increase, at a very high distortion limit.

1 Introduction

Word order differences are among the most important factors determining the performance of statistical machine translation (SMT) on a given language pair (Birch et al., 2009). This is particularly true in the framework of phrase-based SMT (PSMT) (Zens et al., 2002; Koehn et al., 2003; Och and Ney, 2002), an approach that remains highly competitive despite the recent advances of the tree-based approaches.

During the PSMT decoding process, the output sentence is built from left to right, while the input sentence positions can be covered in different orders. Thus, reordering in PSMT can be viewed as the problem of choosing the input permutation that leads to the highest-scoring output sentence. Due to efficiency reasons, however, the input permutation space cannot be fully explored, and is therefore limited with hard reordering constraints.

Although many solutions have been proposed to explicitly model word reordering during decoding, PSMT still largely fails to handle long-range word movements in language pairs with different syntactic structures[1]. We believe this is mostly not due to deficiencies of the existing reordering *models*, but rather to a very coarse definition of the reordering search *space*. Indeed, the existing reordering constraints are rather simple and typically based on word-to-word distances. Moreover, they are uniform throughout the input sentence and insensitive to the actual words being translated. Relaxing this kind of constraints means dramatically increasing the size of the search space and making the reordering model's task extremely complex. As a result, even in language pairs where long reordering is regularly observed, PSMT quality degrades when long word movements are allowed to the decoder.

We address this problem by training a binary classifier to predict whether a given input position should be translated right after another, given the words at those positions and their contexts. When this model is integrated into the decoder, its predic-

[1] For empirical evidence, see for instance (Birch et al., 2009; Galley and Manning, 2008; Bisazza and Federico, 2012).

Transactions of the Association for Computational Linguistics, 1 (2013) 327–340. Action Editor: Philipp Koehn.
Submitted 1/2013; Revised 5/2013; Published 7/2013.

tions can be used not only as an additional feature function, but also as an early indication of whether or not a given reordering path should be further explored. More specifically, at each hypothesis expansion, we consider the set of input positions that are reachable according to the usual reordering constraints, and prune it based only on the reordering model score. Then, the hypothesis can be expanded normally by covering the non-pruned positions. This technique makes it possible to dynamically shape the search space while decoding with a very high distortion limit, which can improve translation quality and efficiency at the same time.

The remainder of the paper is organized as follows. After an overview of the relevant literature, we describe in detail our word reordering model. In the following section, we introduce early pruning of reordering steps as a way to dynamically shape the input permutation space. Finally, we present an empirical analysis of our approach, including intrinsic evaluation of the model and SMT experiments on a well-known Arabic-English news translation task.

2 Previous Work

In this paper, we focus on methods that guide the reordering search *during* the phrase-based decoding process. See for instance (Costa-jussà and Fonollosa, 2009) for a review of pre- and post-reordering approaches that are not treated here.

Assuming a one-to-one correspondence between source and target phrases, reordering in PSMT can be viewed as the problem of searching through a set of permutations of the input sentence. Thus, two sub-problems arise: defining the set of allowed permutations (reordering constraints) and scoring the allowed permutations according to some likelihood criterion (reordering model). We begin with the latter, returning to the constraints later in this section.

2.1 Reordering modeling

In its original formulation, the PSMT approach includes a basic reordering model, called **distortion cost**, that exponentially penalizes longer jumps among consecutively translated phrases ($\tilde{f}_{i-1}, \tilde{f}_i$):

$$d(\tilde{f}_{i-1}, \tilde{f}_i) = e^{-|\text{start}(\tilde{f}_i) - \text{end}(\tilde{f}_{i-1}) - 1|}$$

A number of more sophisticated solutions have been proposed to explicitly model word reordering during decoding. These can mostly be grouped into three families: phrase orientation models, jump models and source decoding sequence models.

Phrase orientation models (Tillmann, 2004; Koehn et al., 2005; Zens and Ney, 2006; Galley and Manning, 2008), also known as lexicalized reordering models, predict the orientation of a phrase with respect to the last translated one, by classifying it as *monotone, swap* or *discontinuous*. These models have proven very useful for short and medium-range reordering and are among the most widely used in PSMT. However, their coarse classification of reordering steps makes them unsuitable to predict long-range reorderings.

Jump models (Al-Onaizan and Papineni, 2006; Green et al., 2010; Yahyaei and Monz, 2010) predict the direction and length of a *jump* to perform after a given input word[2]. Both these works achieve their best Arabic-English results within a rather small DL: namely, 8 in (Al-Onaizan and Papineni, 2006) and 5 in (Green et al., 2010), thus failing to capture the rare but crucial long reorderings that were their main motivation. A drawback of this approach is that long jumps are typically penalized because of their low frequency compared to short jumps. This strong bias is undesirable, given that we are especially interested in detecting probable long reorderings.

Source decoding sequence models predict which input word is likely to be translated at a given state of decoding. For instance, *reordered source language models* (Feng et al., 2010) are smoothed n-gram models trained on a corpus of source sentences reordered to match the target word order. When integrated into the SMT system, they assign a probability to each newly translated word given the n-1 previously translated words. Finally, *source word pair reordering models* (Visweswariah et al., 2011) estimate, for each pair of input words i and j, the cost of translating j right after i given various features of i, j and their respective contexts. Differently from reordered source LMs, these models are discriminative and can profit from richer feature sets. At the same time, they do not employ decoding history-based features, which allows for more effective hy-

[2]In this paper, *input (or source) word* denotes the word at a given position of the input sentence, rather than a word type.

pothesis recombination. The model we are going to present belongs to this last sub-group, which we find especially suitable to predict long reorderings.

2.2 Reordering constraints

The reordering constraint originally included in the PSMT framework and implemented in our reference toolkit, Moses (Koehn et al., 2007), is called **distortion limit (DL)**. This consists in allowing the decoder to skip, or jump, at most k words from the last translated phrase to the next one. More precisely, the limit is imposed on the distortion D between consecutively translated phrases $(\tilde{f}_{i-1}, \tilde{f}_i)$:

$$\mathrm{D}(\tilde{f}_{i-1}, \tilde{f}_i) = \left|\mathrm{start}(\tilde{f}_i) - \mathrm{end}(\tilde{f}_{i-1}) - 1\right| \leq \mathrm{DL}$$

Limiting the input permutation space is necessary for beam-search PSMT decoders to function in linear time. Reordering constraints are also important for translation quality because the existing models are typically not discriminative enough to guide the search over very large sets of reordering hypotheses.

Despite their crucial effects on the complexity of reordering modeling, though, reordering constraints have drawn less attention in the literature. The existing reordering constraints are typically based on word-to-word distances – IBM (Berger et al., 1996) and DL (Koehn et al., 2007) – or on permutation patterns – ITG (Wu, 1997). Both kinds of constraints are uniform throughout the input sentence, and insensitive to the word being translated and to its context. This results in a very coarse definition of the reordering search space, which is problematic in language pairs with different syntactic structures.

To address this problem, Yahyaei and Monz (2010) present a technique to dynamically set the DL: they train a classifier to predict the most probable jump length after each input word, and use the predicted value as the DL after that position. Unfortunately, this method can generate inconsistent constraints leading to decoding dead-ends. As a solution, the dynamic DL is relaxed when needed to reach the first uncovered position. Translation improvements are reported only on a small-scale task with short sentences (BTEC), over a baseline that includes a very simple reordering model. In our work we develop this idea further and use a reordering model to predict which specific input words, rather than input intervals, are likely be translated next. Moreover, our solution is not affected by the constraint inconsistency problem (see Sect. 4).

In another related work, Bisazza and Federico (2012) generate likely reorderings of the input sentence by means of language-specific fuzzy rules based on shallow syntax. Long jumps are then suggested to the PSMT decoder by reducing the distortion cost for specific pairs of input words. In comparison to the dynamic DL, that is a much finer way to define the reordering space, leading to consistent improvements of both translation quality and efficiency over a strong baseline. However, the need of specific reordering rules makes the method harder to apply to new language pairs.

3 The WaW reordering model

We model reordering as the problem of deciding whether a given input word should be translated after another (**W**ord-**a**fter-**W**ord). This formulation is particularly suitable to help the decoder decide whether a reordering path is promising enough to be further explored. Moreover, when translating a sentence, choosing the next source word to translate appears as a more natural problem than guessing how much to the left or to the right we should move from the current source position. The WaW reordering model addresses a binary decision task through the following maximum-entropy classifier:

$$P(R_{i,j}{=}Y|f_1^J, i, j) = \frac{exp[\sum_m \lambda_m h_m(f_1^J, i, j, R_{i,j}{=}Y)]}{\sum_{Y'} exp[\sum_m \lambda_m h_m(f_1^J, i, j, R_{i,j}{=}Y')]}$$

where f_1^J is a source sentence of J words, h_m are feature functions and λ_m the corresponding feature weights. The outcome Y can be either 1 or 0, with $R_{i,j}{=}1$ meaning that the word at position j is translated *right after* the word at position i.

Our WaW reordering model is strongly related to that of Visweswariah et al. (2011) – hereby called Travelling Salesman Problem (TSP) model – with few important differences: (i) we do not include in the features any explicit indication of the jump length, in order to avoid the bias on short jumps; (ii) they train a linear model with MIRA (Crammer and Singer, 2003) by minimizing the number

of input words that get placed after the wrong position, while we use a maximum-entropy classifier trained by maximum-likelihood; (iii) they use an off-the shelf TSP solver to find the best source sentence permutation and apply it as pre-processing to training and test data. By contrast, we integrate the maximum-entropy classifier directly into the SMT decoder and let all its other models (phrase orientation, translation, target LM etc.) contribute to the final reordering decision.

3.1 Features

Like the TSP model (Visweswariah et al., 2011), the WaW model builds on binary features similar to those typically employed for dependency parsing (McDonald et al., 2005): namely, combinations of surface forms or POS tags of the words i and j and their context. Our feature templates are presented in Table 1. The main novelties with respect to the TSP model are the mixed word-POS templates (rows 16-17) and the shallow syntax features. In particular, we use the chunk types of i, j and their context (18-19), as well as the chunk head words of i and j (20). Finally we add a feature to indicate whether the words i and j belong to the same chunk (21). The jump orientation – forward/backward – is included in the features that represent the words comprised between i and j (rows 6, 7, 14, 15). No explicit indication of the jump length is included in any feature.

3.2 Training data

To generate training data for the classifier, we first extract reference reorderings from a word-aligned parallel corpus. Given a parallel sentence, different heuristics may be used to convert arbitrary word alignments to a source permutation (Birch et al., 2010; Feng et al., 2010; Visweswariah et al., 2011). Similarly to this last work, we compute for each source word f_i the mean $\overline{a_i}$ of the target positions aligned to f_i, then sort the source words according to this value.[3] As a difference, though, we do not discard unaligned words but assign them the mean

[3]Using the mean of the aligned indices makes the generation of reference permutations more robust to alignment errors. Admittedly, this heuristic does not handle well the case of source words that are correctly aligned to non-consecutive target words. However, this phenomenon is also not captured by standard PSMT models, who only learn continuous phrases.

	$i-2$	$i-1$	i	$i+1$	b	$j-1$	j	$j+1$
1			w				w	
2		w	w				w	
3	w	w	w				w	
4		w	w				w	w
5			w	w		w	w	
6					w	w	w	
7					w_{all}	w	w	
8			p				p	
9		p	p				p	
10	p	p	p				p	
11		p	p				p	p
12			p	p		p	p	
13		p	p	p		p	p	p
14					p	p	p	
15					p_{all}	p	p	
16			w				p	
17			p				w	
18			c				c	
19		c	c	c		c	c	c
20			h				h	
21	belong_to_same_chunk(i,j)?							

w: word identity, p: POS tag, c: chunk type, h: chunk head

Table 1: Feature templates used to learn whether a source position j is to be translated right after i. Positions comprised between i and j are denoted by b and generate two feature templates: one for each position (6 and 14) and one for the concatentation of them all (7 and 15).

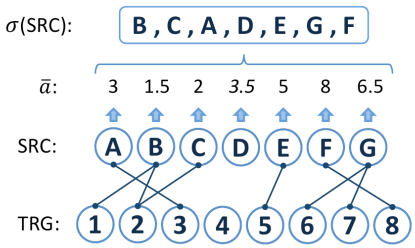
of their neighbouring words' alignment means, so that a complete permutation of the source sentence (σ) is obtained. Table 2(a) illustrates this procedure.

Given the reference permutation, we then generate positive and negative training samples by simulating the decoding process. We traverse the source positions in the order defined by σ, keeping track of the positions that have already been covered and, for each $t : 1 \leq t \leq J$, generate:

- one positive sample ($R_{\sigma_t,\sigma_{t+1}}$=1) for the source position that comes right after it,
- a negative sample ($R_{\sigma_t,u}$=0) for each source position in $\{u : \sigma_t-\delta+1 < u < \sigma_t+\delta+1 \ \wedge \ u \neq \sigma_{t+1}\}$ that has not yet been translated.

Here, the *sampling window* δ serves to control the size of the training data and the proportion between positive and negative samples. Its value naturally correlates with the DL used in decoding. The generation of training samples is illustrated by Table 2(b).

(a) Converting word alignments to a permutation: source words are sorted by their target alignments mean $\bar{a}$. The unaligned word "D" is assigned the mean of its neighbouring words' $\bar{a}$ values $(2 + 5)/2 = 3.5$:



(b) Generating binary samples by simulating the decoding process: shaded rounds represent covered positions, while dashed arrows represent negative samples:

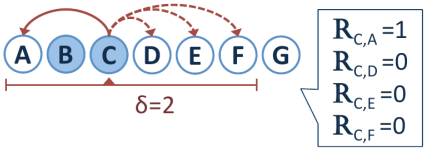


Table 2: The classifier's training data generation process.

3.3 Integration into phrase-based decoding

Rather than using the new reordering model for data pre-processing as done by (Visweswariah et al., 2011), we directly integrate it into the PSMT decoder Moses (Koehn et al., 2007).

Two main computation phases are required by the WaW model: (i) at system initialization time, all feature weights are loaded into memory, and (ii) before translating each new sentence, features are extracted from it and model probabilities are pre-computed for each pair of source positions (i, j) such that $|j - i - 1| \leq$ DL. Note that this efficient solution is possible because our model does not employ decoding history-based features, like the word that was translated before the last one, or like the previous jump legth. This is an important difference with respect to the reordered source LM proposed by Feng et al. (2010), which requires inclusion of the last n translated words in the decoder state.

Fig. 1 illustrates the scoring process: when a partial translation hypothesis $\mathcal{H}$ is expanded by covering a new source phrase $\tilde{f}$, the model returns the log-probability of translating the words of $\tilde{f}$ in that particular order, just after the last translated word of $\mathcal{H}$. In details, this is done by converting the phrase-internal word alignment[4] to a source permutation, in just the same way it was done to produce the model's training examples. Thus, the global score is independent from phrase segmentation, and normalized across outputs of different lengths: that is, the probability of any complete hypothesis decomposes into J factors, where J is the length of the input sentence.

The WaW reordering model is fully compatible with, and complementary to the lexicalized reordering (phrase orientation) models included in Moses.

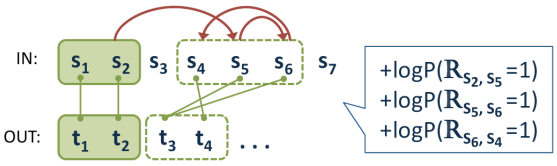


Figure 1: Integrating the binary word reordering model into a phrase-based decoder: when a new phrase is covered (dashed boxes), the model returns the log-probability of translating its words in the order defined by the phrase-internal word alignment.

4 Early pruning of reordering steps

We now explain how the WaW reordering model can be used to dynamically refine the input permutation space. This method is not dependent on the particular classifier described in this paper, but can in principle work with any device estimating the probability of translating a given input word after another.

The method consists of querying the reordering model at the time of hypothesis expansion, and filtering out hypotheses solely based on their reordering score. The rationale is to avoid costly hypothesis expansions for those source positions that the reordering model considers very unlikely to be covered at a given point of decoding. In practice, this works as follows:

- at each hypothesis expansion, we first enumerate the set of uncovered input positions that are reachable within a fixed DL, and query the WaW reordering model for each of them[5];

[4] Phrase-internal alignments are provided in the phrase table.

[5] The score used to prune a new word range $\tilde{f}$ is the log probability of translating the first aligned word of $\tilde{f}$ right after the last translated word of the current hypothesis. See also Sect. 3.3.

- only based on the WaW score, we apply histogram and threshold pruning to this set and proceed to expand the non-pruned positions.

Furthermore, it is possible to ensure that local reorderings are always allowed, by setting a so-called *non-prunable-zone* of width ϑ around the last covered input position.[6]

According to how the DL, pruning parameters, and ϑ are set, we can actually aim at different targets: with a low DL, loose pruning parameters, and ϑ=0 we can try to speed up search without sacrificing much translation quality. With a high DL, strict pruning parameters, and a medium ϑ, we ensure that the standard medium-range reordering space is explored, as well as those few long jumps that are promising according to the reordering model. In our experiments, we explore this second option with the setting DL=18 and ϑ=5.

The underlying idea is similar to that of *early pruning* proposed by Moore and Quirk (2007), which consisted in discarding possible extensions of a partial hypothesis based on their estimated score *before* computing the exact language model score. Our technique too has the effect of introducing additional points at which the search space is pruned. However, while theirs was mainly an optimization technique meant to avoid useless LM queries, we instead aim at refining the search space by exploiting the fact that some SMT models are more important than others at different stages of the translation process. Our approach actually involves a continuous alternation of two processes: during hypothesis expansion the reordering score is combined with all other scores, while during early pruning some reordering decisions are taken only based on the reordering score. In this way, we try to combine the benefits of fully integrated reordering models with those of monolingual pre-ordering methods.

5 Evaluation

We test our approach on an Arabic-English news translation task where sentences are typically long and complex. In this language pair, long reordering errors mostly concern verbs, as all of Subject-Verb-Object (SVO), VSO and, more rarerly, VOS constructions are attested in modern written Arabic. This issue is well known in the SMT field and was addressed by several recent works, with deep or shallow parsing-based techniques (Green et al., 2009; Carpuat et al., 2012; Andreas et al., 2011; Bisazza et al., 2012). We question whether our approach – which is not conceived to solve this specific problem, nor requires manual rules to predict verb reordering – will succeed in improving long reordering in a fully data-driven way.

As SMT training data, we use all the in-domain parallel data provided for the NIST-MT09 evaluation for a total of 986K sentence pairs (31M English words).[7] The target LM used to run the main series of experiments is trained on the English side of all available NIST-MT09 parallel data, UN included (147M words). In the large-scale experiments, the LM training data also include the sections of the English Gigaword that best fit to the development data in terms of perplexity: namely, the Agence France-Presse, Xinhua News Agency and Associated Press Worldstream sections (2130M words in total).

For development and test, we use the newswire sections of the NIST benchmarks: dev06-nw, eval08-nw, eval09-nw consisting of 1033, 813, 586 sentences respectively. Each set includes 4 reference translations and the average sentence length is 33 words. To focus the evaluation on problematic reordering, we also consider a subset of eval09-nw containing only sentences where the Arabic main verb is placed before the subject (vs-09: 299 sent.).[8]

As pre-processing, we apply standard tokenization to the English data, while the Arabic data is segmented with AMIRA (Diab et al., 2004) according to the ATB scheme[9]. The same tool also produces POS tagging and shallow syntax annotation.

[6] See Bisazza (2013) for technical details on the integration of word-level pruning with phrase-level hypothesis expansion.

[7] The in-domain parallel data includes all the provided corpora except the UN proceedings, and the non-newswire parts of the small GALE-Y1-Q4 corpus (that is 9K sentences of audio transcripts and web data). As reported by Green et al. (2010) the removal of UN data does not affect baseline performances on the news benchmarks.

[8] Automatically detected by means of shallow syntax rules.

[9] The Arabic Treebank tokenization scheme isolates conjunctions *w+* and *f+*, prepositions *l+, k+, b+*, future marker *s+*, pronominal suffixes, but not the article *Al+*.

5.1 Reordering model intrinsic evaluation

Before proceeding to the SMT experiments, we evaluate the performance of the WaW reordering model in isolation. All the tested configurations are trained with the freely available MegaM Toolkit[10], implementing the conjugate gradient method (Hestenes and Stiefel, 1952), in maximum 100 iterations. Training samples are generated within a sampling window of width δ=10, from a subset (30K sentences) of the parallel data described above, resulting in 8M training word pairs[11]. Test samples are generated from TIDES-MT04 (1324 sentences, 370K samples with δ=10), one of the corpora included in our SMT training data. Features with less than 20 occurrences are ignored.

Classification accuracy. Table 3 presents precision, recall, and F-score achieved by different feature subsets, where W stands for word-based, P for POS-based and C for chunk-based feature templates. We can see that all feature types contribute to improve the classifier's performance. The word-based model achieves the highest precision but a very low recall, while the POS-based has much more balanced scores. A better performance overall is obtained by combining word-, POS- and mixed word-POS-based features (62.6% F-score). Finally, the addition of chunk-based features yields a further improvement of about 1 point, reaching 63.8% F-score. Given these results, we decide to use the W,P,C model for the rest of the evaluation.

Features (templates)		P	R	F
W	[1-7]	73.1	16.4	26.8
P	[8-15]	69.5	54.8	61.3
W,P	[1-17]	70.2	56.5	62.6
W,P,C	**[1-21]**	**70.6**	**58.1**	**63.8**

Table 3: Classification accuracy of the WaW reordering model on TIDES-MT04, using different feature subsets. The template numbers refer to the rows of Table 1.

Ranking accuracy. A more important aspect to evaluate for our application is how well our model's scores can *rank* a typical set of reordering options. In fact, the WaW model is not meant to be used as a stand-alone classifier, but as one of several SMT feature functions. Moreover, for early reordering pruning to be effective, it is especially important that the correct reordering option be ranked in the top n among those available at the time of a given hypothesis expansion. In order to measure this, we simulate the decoding process by traversing the source words in target order and, for each of them, we examine the ranking of all words that may be translated next (i. e. the uncovered positions within a given DL). We check how often the correct jump was ranked first (Top-1) or at most third (Top-3). We also compute the latter score on long reorderings only (Top-3-long): i. e. backward jumps with distortion D>7 and forward jumps with D>6. In Table 4, results are compared with the ranking produced by standard distortion, which always favors shorter jumps. Two conditions are considered: DL=10 corresponding to the sampling window δ used to produce the training data, and DL=18 that is the maximum distortion of jumps that will be considered in our early-pruning SMT experiment.

Model	DL	DL-err	Top-1	Top-3	Top-3-long back	Top-3-long forw.
Distortion	10	2.4	61.8	79.6	50.7	66.0
	18	0.8	62.0	80.0	18.9	52.3
WaW	10	2.4	71.2	91.2	76.4	69.3
	18	0.8	71.2	91.8	68.0	51.8

Table 4: Word-to-word jump ranking accuracy (%) of standard distortion and WaW reordering model, in different DL conditions. DL-err is the percentage of correct jumps beyond DL. The test set consists of 40K reordering decisions: one for each source word in TIDES-MT04.

We can see that, in terms of overall accuracies, the WaW reordering model outperforms standard distortion by a large margin (about 10% absolute). This is an important result, considering that the jump length, strongly correlating with the jump likelihood, is not directly known to our model. As regards the DL, the higher limit naturally results in a lower DL-error rate (percentage of correct jumps beyond DL): namely 0.8% instead of 2.4%. However, jump prediction becomes much harder: Top-3 accuracy of long jumps by distortion drops from 50.7% to 18.9% (backward) and from 66.0% to 52.3% (forward). Our model is remarkably robust to this effect on backward jumps, where it achieves 68.0% accu-

[10] http://www.cs.utah.edu/~hal/megam/ (Daumé III, 2004).

[11] This is the maximum number of samples manageable by MegaM. However, even scaling from 4M to 8M was only slightly helpful in our experiments. In the future we plan to test other learning approaches that scale better to large data sets.

racy. Due to the syntactic characteristics of Arabic and English, the typical long reordering pattern consists in (i) skipping a clause-initial Arabic verb, (ii) covering a long subject, then finally (iii) jumping back to translate the verb and (iv) jumping forward to continue translating the rest of the sentence (see Fig. 3 for an example).[12] Deciding when to jump back to cover the verb (iii) is the hardest part of this process, and that is precisely where our model seems more helpful, while distortion always prefers to proceed monotonically achieving a very low accuracy of 18.9%. In the case of long forward jumps (iv), instead, distortion is advantaged as the correct choice typically corresponds to translating the first uncovered position, that is the *shortest* jump available from the last translated word. Even here, our model achieves an accuracy of 51.8%, only slightly lower than that of distortion (52.3%).

In summary, the WaW reordering model significantly outperforms distortion in the ranking of long jumps. In the large majority of cases, it is able to rank a correct long jump in the top 3 reordering options, which suggests that it can be effectively used for early reordering pruning.

5.2 SMT experimental setup

Our SMT systems are built with the Moses toolkit, while word alignment is produced by the Berkeley Aligner (Liang et al., 2006). The baseline decoder includes a phrase translation model, a lexicalized reordering model, a 6-gram target language model, distortion cost, word and phrase penalties. More specifically, the baseline reordering model is a **hierarchical phrase orientation model** (Tillmann, 2004; Koehn et al., 2005; Galley and Manning, 2008) trained on all the available parallel data. This variant was shown to outperform the default word-based on an Arabic-English task. To make our baseline even more competitive, we apply **early distortion cost**, as proposed by Moore and Quirk (2007). This function has the same value as the standard one over a complete translation hypothesis, but it anticipates the gradual accumulation of the cost, making hypotheses of the same length more comparable to one another. Note that this option has no effect on the distortion limit, but only on the distortion cost *feature function*. As proposed by Johnson et al. (2007), statistically improbable phrase pairs are removed from the translation model. The language models are estimated by the IRSTLM toolkit (Federico et al., 2008) with modified Kneser-Ney smoothing (Chen and Goodman, 1999).

Feature weights are optimized by minimum BLEU-error training (Och, 2003) on dev06-nw. To reduce the effects of the optimizer instability, we tune each configuration four times and use the average of the resulting weight vectors to translate the test sets, as suggested by Cettolo et al. (2011).

Finally, eval08-nw is used to select the early pruning parameters for the last experiment, while eval09-nw is always reserved as blind test.

5.3 Evaluation metrics

We evaluate global translation quality with **BLEU** (Papineni et al., 2002) and **METEOR** (Banerjee and Lavie, 2005). These metrics, though, are only indirectly sensitive to word order, and especially unlikely to capture improvements at the level of long-range reordering. For this reason, we also compute the Kendall Reordering Score or **KRS** (Birch et al., 2010) which is a positive score based on the Kendall's Tau distance between the source-output permutation π and the source-reference permutations σ:

$$\text{KRS}(\pi, \sigma) = (1 - \sqrt{K(\pi, \sigma)}) \cdot \text{BP}$$

$$K(\pi, \sigma) = \frac{\sum_{i=1}^{n} \sum_{j=1}^{n} \mathbf{d}(i, j)}{\frac{1}{2} n(n-1)}$$

$$\mathbf{d}(i, j) = \begin{cases} 1 & \text{if } \pi_i < \pi_j \text{ and } \sigma_i > \sigma_j \\ 0 & \text{otherwise} \end{cases}$$

where BP is a sentence-level brevity penalty, similar to that of BLEU. The KRS is robust to lexical choice because it performs no comparison between output and reference *words*, but only between the *positions* of their translations. Besides, it was shown to correlate strongly with human judgements of fluency.

Our work specifically addresses long-range reordering phenomena in language pairs where these are quite rare, although crucial for preserving the source text meaning. Hence, an improvement at this level may not be detected by the general-purpose metrics. We then develop a KRS variant that is only

[12] Clearly, we would expect different figures from testing the model on another language pair like German-English, where the verb is often postponed in the source with respect to the target.

sensitive to the positioning of specific input words. Assuming that each input word f_i is assigned a weight λ_i, the formula above is modified as follows:

$$\mathbf{d}_\lambda(i,j) = \begin{cases} \lambda_i+\lambda_j & \text{if } \pi_i < \pi_j \text{ and } \sigma_i > \sigma_j \\ 0 & \text{otherwise} \end{cases}$$

A similar element-weighted version of Kendall Tau was proposed by Kumar and Vassilvitskii (2010) to evaluate document rankings in information retrieval. Because long reordering errors in Arabic-English mostly affect verbs, we set the weights to 1 for verbs and 0 for all other words to only capture verb reordering errors, and call the resulting metric **KRS-V**.

The source-reference word alignments needed to compute the reordering scores are generated by the Berkeley Aligner previously trained on the training data. Source-output word alignments are instead obtained from the decoder's trace.

5.4 Results and discussion

To motivate the choice of our baseline setup (early distortion cost and DL=8), we first compare the performance of *standard* and *early* distortion costs under various DL conditions.

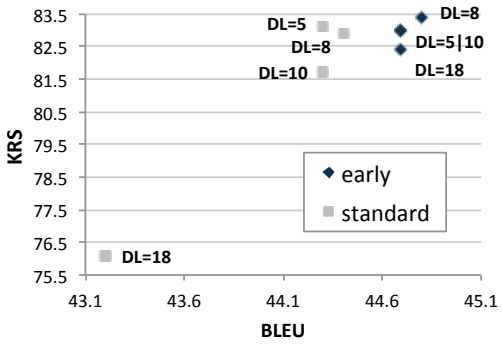


Figure 2: Standard *vs* early distortion cost results on eval08-nw under different distortion limits (DL), using the medium-size LM. Best scores are on top-right corner.

As shown in Fig. 2, most results are close to each other in terms of BLEU and KRS, but early distortion consistently outperforms the standard one (statistically significant). The most striking difference appears at a very high distortion limit (18), where standard distortion scores drop by more than 1 BLEU point and almost 7 KRS points! Early distortion is much more robust (only -1 KRS when going from DL=8 to DL=18), which makes our baseline system especially strong at the level of reordering.

Table 5 presents the results obtained by integrating the WaW reordering model as an additional feature function, and by applying early reordering pruning. The upper part of the table refers to the medium-scale evaluation, while the lower part refers to the large-scale evaluation. In each part, statistical significance is computed against the baseline [**B**] by approximate randomization as in (Riezler and Maxwell, 2005). Run times are obtained by an Intel Xeon X5650 processor on the first 500 sentences of eval08-nw, and exclude loading time of all models.

Medium-scale evaluation. Integrating the WaW model as an additional feature function results in small but consistent improvements in all DL conditions, which shows that this type of model conveys information that is missing from the state-of-the-art reordering models. As regards efficiency, the new model makes decoding time increase by 8%.

Among the DL settings considered, DL=8 is confirmed as the optimal one – with or without WaW model. Raising the DL to 18 with no special pruning has a negative impact on both translation quality and efficiency. The effect is especially visible on the reordering scores: that is, from 84.7 to 83.9 KRS and from 86.2 to 85.8 KRS-V on eval09-nw. Run times are almost doubled: from 87 to 164 and from 94 to 178 ms/word, that is a 89% increase.

We then proceed to the last experiment where the reordering space is dynamically pruned based on the WaW model score. As explained in Sect. 4, a non-prunable-zone of width ϑ=5 is set around the last covered position. To set the early pruning parameters, we perform a grid search over the values (1, 2, 3, 4, 5) for histogram and (0.5, 0.25, 0.1) for relative threshold, and select the values that achieve the best BLEU and KRS on eval08-nw, namely 3 (histogram) and 0.1 (threshold). The resulting configuration is then re-optimized by MERT on dev06-nw. This setting implies that, at a given point of decoding where i is the last covered position, a new word can be translated only if:

- it lies within a DL of 5 from i, or
- it lies within a DL of 18 from i and its WaW reordering score is among the top 3 and at least equal to $1/10$ of the best score (in linear space).

As shown in Table 5, early pruning achieves the best results overall: despite the high DL, we report

DL	Reo.models	eval08-nw				eval09-nw				vs-09	ms/
		bleu	**met**	**krs**	**krs-V**	**bleu**	**met**	**krs**	**krs-V**	**krs-V**	**word**
Using the medium-size LM (147M English tokens):											
5	hier.lexreo, early disto	44.7	35.1$^{\blacktriangledown}$	83.0$^{\blacktriangledown}$	84.7$^{\blacktriangledown}$	50.3$^{\triangledown}$	38.1	84.6	85.9	84.7	59
	+ WaW model	44.8	35.1	83.7	85.4	51.0$^{\blacktriangle}$	38.3$^{\blacktriangle}$	85.1$^{\blacktriangle}$	86.6$^{\triangle}$	85.5$^{\blacktriangle}$	64
8	hier.lexreo, early disto[**B**]	44.8	35.2	83.4	85.6	50.6	38.1	84.7	86.2	84.8	87
	+ WaW model	45.0	35.2	83.7$^{\triangle}$	85.9	51.1$^{\blacktriangle}$	38.3$^{\blacktriangle}$	85.1$^{\blacktriangle}$	86.8$^{\blacktriangle}$	85.8$^{\blacktriangle}$	94
18	hier.lexreo, early disto	44.7	34.9$^{\blacktriangledown}$	82.4$^{\blacktriangledown}$	84.9$^{\blacktriangledown}$	50.3	38.0$^{\triangledown}$	83.9$^{\blacktriangledown}$	85.8$^{\triangledown}$	84.3$^{\triangledown}$	164
	+ WaW model	44.8	35.2	82.7$^{\blacktriangledown}$	85.5	51.0$^{\triangle}$	38.3$^{\blacktriangle}$	84.2$^{\triangledown}$	86.2	85.2	178
	+ early reo.pruning(ϑ=5)	45.0	35.3	83.7$^{\triangle}$	**86.3**$^{\blacktriangle}$	50.9	38.3$^{\blacktriangle}$	84.9	**87.0**$^{\blacktriangle}$	**86.2**$^{\blacktriangle}$	**68**
Using the large interpolated LM (2130M English tokens) and double beam-size:											
8	hier.lexreo, early disto[**B**]	46.3	35.0	83.2	85.0	51.6	38.3	84.5	85.8	84.5	2579
18	hier.lexreo, early disto	45.9$^{\triangledown}$	34.9$^{\blacktriangledown}$	81.7$^{\blacktriangledown}$	84.1$^{\blacktriangledown}$	51.4	38.1$^{\blacktriangledown}$	83.0$^{\blacktriangledown}$	84.6$^{\blacktriangledown}$	83.1$^{\blacktriangledown}$	5462
	+WaW+reo.pruning(ϑ=5)	46.3	35.2	83.4	**85.7**$^{\blacktriangle}$	52.8$^{\blacktriangle}$	38.6$^{\blacktriangle}$	84.6	**86.6**$^{\blacktriangle}$	**85.5**$^{\blacktriangle}$	**1588**

Table 5: Effects of WaW reordering modeling and early reordering pruning on translation quality, measured with % BLEU, METEOR, and Kendall Reordering Score: regular (KRS) and verb-specific (KRS-V). Statistically significant differences with respect to the baseline [**B**] are marked with $^{\blacktriangle\blacktriangledown}$ at the $p \leq .05$ level and $^{\triangle\triangledown}$ at the $p \leq .10$ level. Decoding time is measured in milliseconds per input word.

no loss in BLEU, METEOR and KRS, but we actually see several improvements. In particular, the gains on the blind test eval09-nw are +0.3 BLEU, +0.2 METEOR and +0.2 KRS (only METEOR is significant). While these gains are admittedly small, we recall that our techniques affect rare and isolated events which can hardly emerge from the general-purpose evaluation metrics. Moreover, to our knowledge, this is the first time that a PSMT system is shown to maintain a good performance on this language pair while admitting very long-range reorderings.

Finally and more importantly, the reordering of verbs improves significantly on both generic tests and on the VS- sentence subset (vs-09): namely, in the latter, we achieve a notable gain of 1.4 KRS-V.

Efficiency is also largely improved by our early reordering pruning technique: decoding time is reduced to 68 ms/word, corresponding to a 22% speed-up over the baseline.

Large-scale evaluation. We also investigate whether our methods can be useful in a scenario where efficiency is less important and more data is available for training. To this end, we build a very large LM by interpolating the main LM with three other LMs trained on different Gigaword sections (see Sect. 5). Moreover, we relax the decoder's beam size from the default value of 200 to 400 hypotheses, to reduce the risk of search errors and obtain the best possible baseline performance.

By comparing the large-scale with the medium-scale baseline in Table 5, we note that the addition of LM data is especially beneficial for BLEU (+1.5 on eval08-nw and +1.0 on eval09-nw), but not as much for the other metrics, which challenges the commonly held idea that more data always improves translation quality.

Here too, relaxing the DL without special pruning hurts not only efficiency but also translation quality: all the scores decrease considerably, showing that even the stronger LM is not sufficient to guide search through a very large reordering search space.

As for our enhanced system, it achieves similar gains as in the medium-scale scenario: that is, BLEU and METEOR are preserved or slightly improved despite the very high DL, while all the reordering scores increase. In particular, we report statistically significant improvements in the reordering of verbs, which is where the impact of our method is expected to concentrate (+0.7, +0.8 and +1.0 KRS-V on eval08-nw, eval09-nw and vs-09, respectively).

These results confirm the usefulness of our method not only as an optimization technique, but also as a way to improve translation quality on top of a very strong baseline.

يواصل سفير المملكة العربية السعودية لدى لبنان عبدالعزيز خوجة تحرك ـه في اتجاه ...

SRC	verb	subj.											obj.		compl.
	ywASl	sfyr	Almmlkp	AlErbyp	AlsEwdyp	ldY	lbnAn	EbdAlEzyz	xwjp				tHrk	-h	fy AtjAh ...
	continues	*ambassador*	*Kingdom*	*Arabian*	*Saudi*	*to*	*Lebanon*	*Abdulaziz*	*Khawja*				*move his*		*in direction*
REF	The Kingdom of Saudi Arabia 's ambassador to Lebanon Abdulaziz Khawja **continues** his moves towards ...														
BASE	**continue** to Saudi Arabian ambassador to Lebanon , Abdulaziz Khwja its move in the direction of ...														
NEW	The Kingdom of Saudi Arabia 's ambassador to Lebanon , Abdulaziz Khwja **continue** its move in the direction of ...														

فيما دعا ـهم رئيس المكتب السياسي لـ حركة حماس خالد مشعل الى التزام الحياد

SRC	adv.	verb	obj.	subj.								compl.			
	fymA	dEA	-hm	r}ys	Almktb	AlsyAsy	l-	Hrkp	HmAs	xAld	m$El	AlY	AltzAm	AlHyAd	
	meanwhile	*called*	*them*	*head*	*bureau*	*political*	*of*	*movement*	*Hamas*	*Khaled*	*Mashal*	*to*	*necessity*	*neutrality*	
REF	Meanwhile, the Head of the Political Bureau of the Hamas movement, Khaled Mashal, **called upon them** to remain neutral														
BASE	The **called them**, head of Hamas' political bureau, Khalid Mashal, to remain neutral														
NEW	The head of Hamas' political bureau, Khalid Mashal, **called on them** to remain neutral														

Figure 3: Long reordering examples showing improvements over the baseline system (BASE) when the DL is raised to 18 and early pruning based on WaW reordering scores is enabled (NEW).

Long jumps statistics and examples. To better understand the behavior of the early-pruning system, we extract phrase-to-phrase jump statistics from the decoder log file. We find that 132 jumps beyond the non-prunable zone (D>5) were performed to translate the 586 sentences of eval09-nw; 38 out of these were longer than 8 and mostly concentrated on the VS- sentence subset (27 jumps D>8 performed in vs-09).[13] This and the higher reordering scores suggest that long jumps are mainly carried out to correctly reorder clause-inital verbs over long subjects.

Fig. 3 shows two Arabic sentences taken from eval09-nw, that were erroneuously reordered by the baseline system. The system including the WaW model and early reordering pruning, instead, produced the correct translation. The first sentence is a typical example of VSO order with a long subject: while the baseline system left the verb in its Arabic position, producing an incomprehensible translation, the new system placed it rightly between the English subject and object. This reordering involved two long jumps: one with D=9 backward and one with D=8 forward.

The second sentence displays another, less common, Arabic construction: namely VOS, with a personal pronoun object. In this case, a backward jump with D=10 and a forward jump with D=8 were necessary to achieve the correct reordering.

[13] Statistics computed on the medium-LM system.

6 Conclusions

We have trained a discriminative model to predict likely reordering steps in a way that is complementary to state-of-the-art PSMT reordering models. We have effectively integrated it into a PSMT decoder as additional feature, ensuring that its total score over a complete translation hypothesis is consistent across different phrase segmentations. Lastly, we have proposed early reordering pruning as a novel method to dynamically shape the input reordering space and capture long-range reordering phenomena that are often critical when translating between languages with different syntactic structures.

Evaluated on a popular Arabic-English news translation task against a strong baseline, our approach leads to similar or even higher BLEU, METEOR and KRS scores at a very high distortion limit (18), which is by itself an important achievement. At the same time, the reordering of verbs, measured with a novel version of the KRS, is consistently improved, while decoding gets significantly faster. The improvements are also confirmed when a very large LM is used and the decoder's beam size is doubled, which shows that our method reduces not only search errors but also model errors even when baseline models are very strong.

Word reordering is probably the most difficult aspect of SMT and an important factor of both its quality and efficiency. Given its strong interaction with the other aspects of SMT, it appears natural to solve

word reordering during decoding, rather than before or after it. To date, however, this objective was only partially achieved. We believe there is a promising way to go between fully-integrated reordering models and monolingual pre-ordering methods. This work has started to explore it.

Aknowledgments

This work was partially funded by the European Union FP7 grant agreement 287658 (EU-BRIDGE).

References

Yaser Al-Onaizan and Kishore Papineni. 2006. Distortion models for statistical machine translation. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*, pages 529–536, Sydney, Australia, July.

Jacob Andreas, Nizar Habash, and Owen Rambow. 2011. Fuzzy syntactic reordering for phrase-based statistical machine translation. In *Proceedings of the Sixth Workshop on Statistical Machine Translation*, pages 227–236, Edinburgh, Scotland, July.

Satanjeev Banerjee and Alon Lavie. 2005. METEOR: An automatic metric for MT evaluation with improved correlation with human judgments. In *Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*, pages 65–72, Ann Arbor, Michigan, June.

A. L. Berger, P. F. Brown, S. A. Della Pietra, V. J. Della Pietra, J. R. Gillett, A. S. Kehler, and R. L. Mercer. 1996. Language translation apparatus and method of using context-based translation models. United States Patent, No. 5510981, Apr.

Alexandra Birch, Phil Blunsom, and Miles Osborne. 2009. A quantitative analysis of reordering phenomena. In *StatMT '09: Proceedings of the Fourth Workshop on Statistical Machine Translation*, pages 197–205, Morristown, NJ, USA.

Alexandra Birch, Miles Osborne, and Phil Blunsom. 2010. Metrics for MT evaluation: evaluating reordering. *Machine Translation*, 24(1):15–26.

Arianna Bisazza and Marcello Federico. 2012. Modified distortion matrices for phrase-based statistical machine translation. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 478–487, Jeju Island, Korea, July.

Arianna Bisazza, Daniele Pighin, and Marcello Federico. 2012. Chunk-lattices for verb reordering in Arabic-English statistical machine translation. *Machine Translation, Special Issue on MT for Arabic*, 26(1-2):85–103.

Arianna Bisazza. 2013. *Linguistically Motivated Reordering Modeling for Phrase-Based Statistical Machine Translation*. Ph.D. thesis, University of Trento. http://eprints-phd.biblio.unitn.it/1019/.

Marine Carpuat, Yuval Marton, and Nizar Habash. 2012. Improved Arabic-to-English statistical machine translation by reordering post-verbal subjects for word alignment. *Machine Translation, Special Issue on MT for Arabic*, 26(1-2):105–120.

Mauro Cettolo, Nicola Bertoldi, and Marcello Federico. 2011. Methods for smoothing the optimizer instability in SMT. In *MT Summit XIII: the Thirteenth Machine Translation Summit*, pages 32–39, Xiamen, China.

Stanley F. Chen and Joshua Goodman. 1999. An empirical study of smoothing techniques for language modeling. *Computer Speech and Language*, 4(13):359–393.

Marta R. Costa-jussà and José A. R. Fonollosa. 2009. State-of-the-art word reordering approaches in statistical machine translation: A survey. *IEICE TRANSACTIONS on Information and Systems*, E92-D(11):2179–2185.

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

Hal Daumé III. 2004. Notes on CG and LM-BFGS optimization of logistic regression. Paper available at `http://pub.hal3.name`, implementation available at `http://hal3.name/megam`.

Mona Diab, Kadri Hacioglu, and Daniel Jurafsky. 2004. Automatic Tagging of Arabic Text: From Raw Text to Base Phrase Chunks. In Daniel Marcu Susan Dumais and Salim Roukos, editors, *HLT-NAACL 2004: Short Papers*, pages 149–152, Boston, Massachusetts, USA.

Marcello Federico, Nicola Bertoldi, and Mauro Cettolo. 2008. IRSTLM: an Open Source Toolkit for Handling Large Scale Language Models. In *Proceedings of Interspeech*, pages 1618–1621, Melbourne, Australia.

Minwei Feng, Arne Mauser, and Hermann Ney. 2010. A source-side decoding sequence model for statistical machine translation. In *Conference of the Association for Machine Translation in the Americas (AMTA)*, Denver, Colorado, USA.

Michel Galley and Christopher D. Manning. 2008. A simple and effective hierarchical phrase reordering model. In *EMNLP '08: Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 848–856, Morristown, NJ, USA.

Spence Green, Conal Sathi, and Christopher D. Manning. 2009. NP subject detection in verb-initial Arabic clauses. In *Proceedings of the Third Workshop*

on Computational Approaches to Arabic Script-based Languages (CAASL3), Ottawa, Canada.

Spence Green, Michel Galley, and Christopher D. Manning. 2010. Improved models of distortion cost for statistical machine translation. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*, pages 867–875, Los Angeles, California.

Magnus R. Hestenes and Eduard Stiefel. 1952. Methods of conjugate gradients for solving linear systems. *Journal of Research of the National Bureau of Standards*, 49(6):409–436.

H. Johnson, J. Martin, G. Foster, and R. Kuhn. 2007. Improving translation quality by discarding most of the phrasetable. In *In Proceedings of EMNLP-CoNLL 07*, pages 967–975.

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

Philipp Koehn, Amittai Axelrod, Alexandra Birch Mayne, Chris Callison-Burch, Miles Osborne, and David Talbot. 2005. Edinburgh system description for the 2005 IWSLT speech translation evaluation. In *Proc. of the International Workshop on Spoken Language Translation*, October.

P. Koehn, H. Hoang, A. Birch, C. Callison-Burch, M. Federico, N. Bertoldi, B. Cowan, W. Shen, C. Moran, R. Zens, C. Dyer, O. Bojar, A. Constantin, and E. Herbst. 2007. Moses: Open Source Toolkit for Statistical Machine Translation. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics Companion Volume Proceedings of the Demo and Poster Sessions*, pages 177–180, Prague, Czech Republic.

Ravi Kumar and Sergei Vassilvitskii. 2010. Generalized distances between rankings. In *Proceedings of the 19th international conference on World Wide Web*, pages 571–580, New York, NY, USA. ACM.

Percy Liang, Ben Taskar, and Dan Klein. 2006. Alignment by agreement. In *Proceedings of the Human Language Technology Conference of the NAACL, Main Conference*, pages 104–111, New York City, USA, June.

Ryan McDonald, Fernando Pereira, Kiril Ribarov, and Jan Hajič. 2005. Non-projective dependency parsing using spanning tree algorithms. In *Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing*, HLT '05, pages 523–530, Stroudsburg, PA, USA.

Robert C. Moore and Chris Quirk. 2007. Faster beam-search decoding for phrasal statistical machine translation. In *In Proceedings of MT Summit XI*, pages 321–327, Copenhagen, Denmark.

F. Och and H. Ney. 2002. Discriminative training and maximum entropy models for statistical machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 295–302, Philadelhpia, PA.

Franz Josef Och. 2003. Minimum Error Rate Training in Statistical Machine Translation. In Erhard Hinrichs and Dan Roth, editors, *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics*, pages 160–167.

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 of the Association of Computational Linguistics (ACL)*, pages 311–318, Philadelphia, PA.

Stefan Riezler and John T. Maxwell. 2005. On some pitfalls in automatic evaluation and significance testing for MT. In *Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*, pages 57–64, Ann Arbor, Michigan, June.

Christoph Tillmann. 2004. A Unigram Orientation Model for Statistical Machine Translation. In *Proceedings of the Joint Conference on Human Language Technologies and the Annual Meeting of the North American Chapter of the Association of Computational Linguistics (HLT-NAACL)*.

Karthik Visweswariah, Rajakrishnan Rajkumar, Ankur Gandhe, Ananthakrishnan Ramanathan, and Jiri Navratil. 2011. A word reordering model for improved machine translation. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 486–496, Edinburgh, Scotland, UK., July.

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

Sirvan Yahyaei and Christof Monz. 2010. Dynamic distortion in a discriminative reordering model for statistical machine translation. In *International Workshop on Spoken Language Translation (IWSLT)*, Paris, France.

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

R. Zens, F. J. Och, and H. Ney. 2002. Phrase-based statistical machine translation. In *25th German Conference on Artificial Intelligence (KI2002)*, pages 18–32, Aachen, Germany. Springer Verlag.