Modified Distortion Matrices for Phrase-Based Statistical Machine Translation

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

Abstract

This paper presents a novel method to suggest long word reorderings to a phrase-based SMT decoder. We address language pairs where long reordering concentrates on few patterns, and use fuzzy chunk-based rules to predict likely reorderings for these phenomena. Then we use reordered n-gram LMs to rank the resulting permutations and select the n-best for translation. Finally we encode these reorderings by modifying selected entries of the distortion cost matrix, on a per-sentence basis. In this way, we expand the search space by a much finer degree than if we simply raised the distortion limit. The proposed techniques are tested on Arabic-English and German-English using well-known SMT benchmarks.

1 Introduction

Despite the large research effort devoted to the modeling of word reordering, this remains one of the main obstacles to the development of accurate SMT systems for many language pairs. On one hand, the phrase-based approach (PSMT) (Och, 2002; Zens et al., 2002; Koehn et al., 2003), with its shallow and loose modeling of linguistic equivalences, appears as the most competitive choice for closely related language pairs with similar clause structures, both in terms of quality and of efficiency. On the other, tree-based approaches (Wu, 1997; Yamada, 2002; Chiang, 2005) gain advantage, at the cost of higher complexity and isomorphism assumptions, on language pairs with radically different word orders.

Lying between these two extremes are language pairs where most of the reordering happens locally, and where long reorderings can be isolated and described by a handful of linguistic rules. Notable examples are the family-unrelated Arabic-English and the related German-English language pairs. Interestingly, on these pairs, PSMT generally prevails over tree-based SMT[1], producing overall high-quality outputs and isolated but critical reordering errors that undermine the global sentence meaning.

Previous works on this type of language pairs have mostly focused on source reordering prior to translation (Xia and McCord, 2004; Collins et al., 2005), or on sophisticated reordering models integrated into decoding (Koehn et al., 2005; Al-Onaizan and Papineni, 2006), achieving mixed results. To merge the best of both approaches – namely, access to rich context in the former and natural coupling of reordering and translation decisions in the latter – we introduce *modified distortion matrices*: a novel method to seamlessly provide to the decoder a set of likely long reorderings pre-computed for a given input sentence. Added to the usual space of local permutations defined by a low distortion limit (DL), this results in a linguistically informed definition of the search space that simplifies the task of the in-decoder reordering model, besides decreasing its complexity.

The paper is organized as follows. After reviewing a selection of relevant works, we analyze salient reordering patterns in Arabic-English and German-English, and describe the corresponding chunk-based reordering rule sets. In the following sections we present a reordering selection technique based on

[1] A good comparison of phrase-based and tree-based approaches across language pairs with different reordering levels can be found in (Zollmann et al., 2008).

Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics, pages 478–487,
Jeju, Republic of Korea, 8-14 July 2012.

reordered n-gram LMs and, finally, explain the notion of modified distortion matrices. In the last part of the paper, we evaluate the proposed techniques on two popular MT tasks.

2 Previous work

Pre-processing approaches to word reordering aim at permuting input words in a way that minimizes the reordering needed for translation: *deterministic reordering* aims at finding a single optimal reordering for each input sentence, which is then translated monotonically (Xia and McCord, 2004) or with a low DL (Collins et al., 2005; Habash, 2007); *non-deterministic reordering* encodes multiple alternative reorderings into a word lattice and lets a monotonic decoder find the best path according to its models (Zhang et al., 2007; Crego and Habash, 2008; Elming and Habash, 2009; Niehues and Kolss, 2009). The latter approaches are ideally conceived as alternative to in-decoding reordering, and therefore require an exhaustive reordering rule set. Two recent works (Bisazza and Federico, 2010; Andreas et al., 2011) opt instead for a *hybrid way*: rules are used to generate multiple likely reorderings, but only for a specific phenomenon – namely verb-initial clauses in Arabic. This yields sparse reordering lattices that can be translated with a regular decoder performing additional reordering.

Reordering rules for pre-processing are either manually written (Collins et al., 2005) or automatically learned from syntactic parses (Xia and McCord, 2004; Habash, 2007; Elming and Habash, 2009), shallow syntax chunks (Zhang et al., 2007; Crego and Habash, 2008) or part-of-speech labels (Niehues and Kolss, 2009). Similarly to hybrid approaches, in this work we use few linguistically informed rules to generate multiple reorderings for selected phenomena but, as a difference, we do not employ lattices to represent them. We also include a competitive in-decoding reordering model in all the systems used to evaluate our methods.

Another large body of work is devoted to the modeling of reordering decisions inside decoding, based on a decomposition of the problem into a sequence of basic reordering steps. Existing approaches range from basic linear distortion to more complex models that are conditioned on the words being translated.

The *linear distortion model* (Koehn et al., 2003) encourages monotonic translations by penalizing source position jumps proportionally to their length. If used alone, this model is inadequate for language pairs with different word orders. Green et al. (2010) tried to improve it with a future distortion cost estimate. Thus they were able to preserve baseline performance at a very high DL, but not to improve it. *Lexicalized phrase orientation models* (Tillmann, 2004; Koehn et al., 2005; Zens and Ney, 2006; Galley and Manning, 2008) predict the orientation of a phrase with respect to the last translated one. These models are known to well handle local reordering and are widely adopted by the PSMT community. However, they are unsuitable to model long reordering as they classify as "discontinuous" every phrase that does not immediately follow or precede the last translated one. *Lexicalized distortion models* predict the jump from the last translated word to the next one, with a class for each possible jump length (Al-Onaizan and Papineni, 2006), or bin of lengths (Green et al., 2010). These models are conceived to deal with long reordering, but can easily suffer from data sparseness, especially for longer jumps occurring less frequently.

Following a typical sequence modeling approach, Feng et al. (2010) train n-gram language models on source data previously reordered in accordance to the target language translation. This method does not directly model reordering decisions, but rather word sequences produced by them. Despite their high perplexities, reordered LMs yield some improvements when integrated to a PSMT baseline that already includes a discriminative phrase orientation model (Zens and Ney, 2006). In this work we use similar models to rank sets of chunk permutations.

Attempting to improve the reordering space definition, Yahyaei and Monz (2010) train a classifier to guess the most likely jump length at each source position, then use its predictions to dynamically set the DL. Translation improvements are obtained on a simple task with mostly short sentences (BTEC).

Modifying the distortion function, as proposed in this paper, makes it possible to expand the pemutation search space by a much finer degree than varying the DL does.

3 Long reordering patterns

Our study focuses on Arabic-English and German-English: two language pairs characterized by uneven distributions of word-reordering phenomena, with long-range movements concentrating on few patterns. In **Arabic-English**, the internal order of most noun phrases needs to be reversed during translation, which is generally well handled by phrase-internal reordering or local distortion. At the constituent level, instead, Arabic admits both SV(O) and VS(O) orders, the latter causing problematic long reorderings. Common errors due to this issue are the absence of main verb in the English translation, or the placement of the main verb before its own subject. In both cases, adequacy is seriously compromised. In **German-English**, the noun phrase structure is similar between source and target languages. However, at the constituent level, the *verb-second* order of German main clauses conflicts with the rigid SVO structure of English, as does the clause-final verb position of German subordinate clauses. As a further complication, German compound verbs are split apart so that the non-finite element (main verb) can appear long after the inflected auxiliary or modal.

Thanks to sophisticated reordering models, state-of-the-art PSMT systems are generally good at handling *local* reordering phenomena that are not captured by phrase-internal reordering. However, they typically fail to predict long reorderings. We believe this is mainly not the fault of the reordering models, but rather of a *too coarse definition* of the search space. To have a concrete idea, consider that a small change of the DL from 5 to 6 words, in a sentence of 8, makes the number of explorable permutations increase from about *9,000* to *22,000*. Existing models cannot be powerful enough to deal with such a rapidly growing search space.

As a result, decoding at very high DLs is not a good solution for these language pairs. Indeed, decent performances are obtained within a low or medium DL, but this obviously comes at the expense of long reorderings, which are often crucial to preserve the general meaning of a translated sentence. For instance, taking English as the target language, it is precisely the relative positioning of predicate arguments that determines their role, in the absence of case markers. Thus, a wrongly reordered verb with minor impact on automatic scores, can be judged very badly by a human evaluator.

We will now describe two rule sets aimed at capturing these reordering phenomena.

4 Shallow syntax reordering rules

To compute the source reorderings, we use chunk-based rules following Bisazza and Federico (2010). Shallow syntax chunking is indeed a lighter and simpler task compared to full parsing, and it can be used to constrain the number of reorderings in a softer way. While rules based on full parses are generally deterministic, chunk-based rules are non-deterministic or fuzzy, as they generate several permutations for each matching sequence[2]. Besides defining a unique segmentation of the sentence, chunk annotation provides other useful information that can be used by the rules – namely chunk type and POS tags[3].

For **Arabic-English** we apply the rules proposed by Bisazza and Federico (2010) aimed at transforming VS(O) sentences into SV(O). Reorderings are generated by moving each verb chunk (VC), alone or with its following chunk, by 1 to 6 chunks to the right. The maximum movement of each VC is limited to the position of the next VC, so that neighboring verb-reordering sequences may not overlap. This rule set was shown to cover most (99.5%) of the verb reorderings observed in a parallel news corpus, including those where the verb must be moved along with an adverbial or a complement.

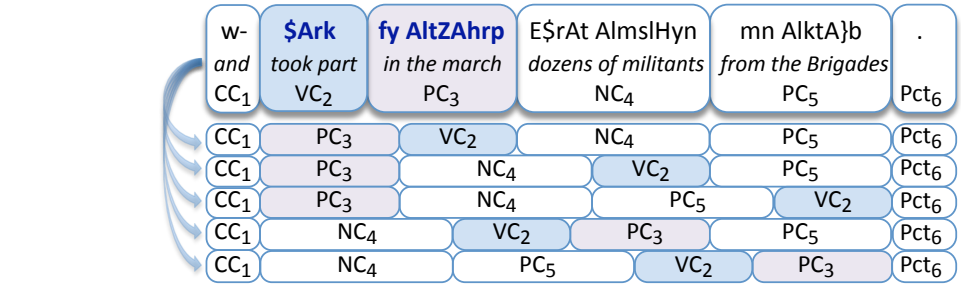
For **German-English** we propose a set of three rules[4] aimed at arranging the German constituents in SVO order:

- *infinitive*: move each infinitive VC right after a preceding punctuation;
- *subordinate*: if a VC is immediately followed by a punctuation, place it after a preceding subordinating conjunction (KOUS) or substitutive relative pronoun (PRELS);

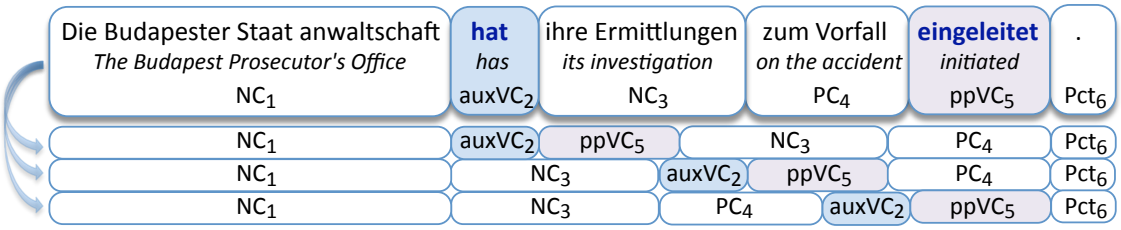
[2]Chunk annotation does not identify subject and complement boundaries, nor the relations among constituents that are needed to deterministically rearrange a sentence in SVO order.

[3]We use AMIRA (Diab et al., 2004) to annotate Arabic and Tree Tagger (Schmid, 1994) to annotate German.

[4]A similar rule set was previously used to produce chunk reordering lattices in (Hardmeier et al., 2010).



(a) Arabic VS(O) clause: five permutations



(b) German *broken* verb chunk: three permutations

Figure 1: Examples of chunk permutations generated by shallow syntax reordering rules. Chunk types: CC conjunction, VC verb (auxiliary/past participle), PC preposition, NC noun, Pct punctuation.

- *broken verb chunk*: join each finite VC (auxiliary or modal) with the nearest following non-finite VC (infinitive or participle). Place the resulting block in any position between the original position of the finite verb and that of the non-finite verb[5].

The application of chunk reordering rules is illustrated by Fig. 1: in the Arabic sentence (a), the subject *'dozens of militants'* is preceded by the main verb *'took part'* and its argument *'to the march'*. The rules generate 5 permutations for one matching sequence (chunks 2 to 5), out of which the 5^{th} is the best for translation. The German sentence (b) contains a broken VC with the inflected auxiliary *'has'* separated from the past participle *'initiated'*. Here, the rules generate 3 permutations for the chunk sequence 2 to 5, corresponding to likely locations of the merged verb phrase, the 1^{st} being optimal.

By construction, both rule sets generate a limited number of permutations per matching sequence: in Arabic at most 12 for each VC; in German at most 3 for each infinitive VC and for each VC-punctuation sequence, at most 10 for each broken VC. Empirically, this yields on average 22 reorderings per sentence in the NIST-MT Arabic benchmark dev06-NW and 3 on the WMT German benchmark test08. Arabic rules are indeed more noisy, which is not surprising as reordering is triggered by any verb chunk.

5 Reordering selection

The number of chunk-based reorderings per sentence varies according to the rule set, to the size of chunks and to the context. A high degree of fuzziness can complicate the decoding process, leaving too much work to the in-decoding reordering model. A solution to this problem is using an external model to score the rule-generated reorderings and discard the less probable ones. In such a way, a further part of reordering complexity is taken out of decoding.

At this end, instead of using a Support Vector Machine classifier as was done in (Bisazza et al., 2011), we apply *reordered n-gram models* that are lighter-weight and more suitable for a ranking task.

Differently from Feng et al. (2010), we train our models on partially reordered data and at the level of chunks. Chunks can be represented simply by their

[5]To bound the number of reorderings, we use the following heuristics. In *'infinitive'* at most 3 punctuations preceding the VC are considered. In *'subordinate'* 1 to 3 chunks are left between the conjunction (or pronoun) and the moved VC to account for the subject. In *'broken VC'* if the distance between the finite and non-finite verb is more than 10 chunks, only the first 5 and last 5 positions of the verb-to-verb span are considered.

type label (such as VC or NC), but also by a combination of the type and head word, to obtain finer lexicalized distributions. LMs trained on different chunk representations can also be applied jointly, by log-linear combination.

We perform reordering selection as follows:

1. Chunk-based reordering rules are applied deterministically to the *source* side of the parallel training data, using word alignment to choose the optimal permutation ("oracle reordering")[6].

2. One or several chunk-level 5-gram LMs are trained on such reordered data, using different chunk representation modes.

3. Reordering rules are applied to the test sentences and the resulting sets of rule-matching sequence permutations are scored by the LMs. The *n*-best reorderings *of each rule-matching sequence* are selected for translation.

In experiments not reported here, we obtained accurate rankings by scoring source permutations with a uniformly weighted combination of two LMs trained on chunk types and on chunk-type+head-word, respectively. In particular, 3-best reorderings of each rule-matching sequence yield reordering recalls of 77.2% in Arabic and 89.3% in German.

6 Modified distortion matrices

We present here a novel technique to encode likely long reorderings of an input sentence, which can be seamlessly integrated into the PSMT framework.

During decoding, the distance between source positions is used for two main purposes: (i) generating a distortion penalty for the current hypothesis and (ii) determining the set of source positions that can be covered at the next hypothesis expansion. We can then tackle the coarseness of both distortion penalty and reordering constraints, by replacing the distance function with a function defined *ad hoc* for each input sentence.

Distortion can be thought of as a matrix assigning a positive integer to any ordered pair of source positions (s_x, s_y). In the linear distortion model this is defined as:

$$D_L(s_x, s_y) = |s_y - s_x - 1|$$

so that moving to the right by 1 position costs 0 and by 2 positions costs 1. Moving to the left by 1 position costs 2 and by 2 positions costs 3, and so on. At the level of phrases, distortion is computed between the last word of the last translated phrase and the first word of the next phrase. We retain this equation as the core distortion function for our model. Then, we modify entries in the matrix such that the distortion cost is minimized for the decoding paths pre-computed with the reordering rules.

Given a source sentence and its set of rule-generated permutations, the linear distortion matrix is modified as follows:

1. non-monotonic jumps (i.e. ordered pairs (s_i, s_{i+1}) such that $s_{i+1} - s_i \neq 1$) are extracted from the permutations;

2. then, for each extracted pair, the corresponding point in the matrix is assigned the lowest possible distortion cost, that is 0 if $s_i < s_{i+1}$ and 2 if $s_i > s_{i+1}$. We call these points *shortcuts*.

Although this technique is approximate and can overgenerate minimal-distortion decoding paths[7], it practically works when the number of encoded permutations per sequence is limited. This makes modifed distortion matrices particularly suitable to encode just those reorderings that are typically missed by phrase-based decoders (see Sect. 3).

Since in this work we use chunk-based rules, we also have to convert chunk-to-chunk jumps into word-to-word shortcuts. We propose two ways to do this, given an ordered pair of chunks (c_x,c_y):

mode A×A : create a shortcut from each word of c_x to each word of c_y;

mode L×F : create only one shortcut from the last word of c_x to the first of c_y.

The former solution admits more chunk-internal permutations with the same minimal distortion cost, whereas the latter implies that the first word of a reordered chunk is covered first and the last is covered last.

[6] Following Bisazza and Federico (2010), the optimal reordering for a source sentence is the one that minimizes distortion in the word alignment to a target translation, measured by number of swaps and sum of distortion costs.

[7] In fact, any decoding path that includes a jump marked as shortcut benefits from the same distortion discount in that point.

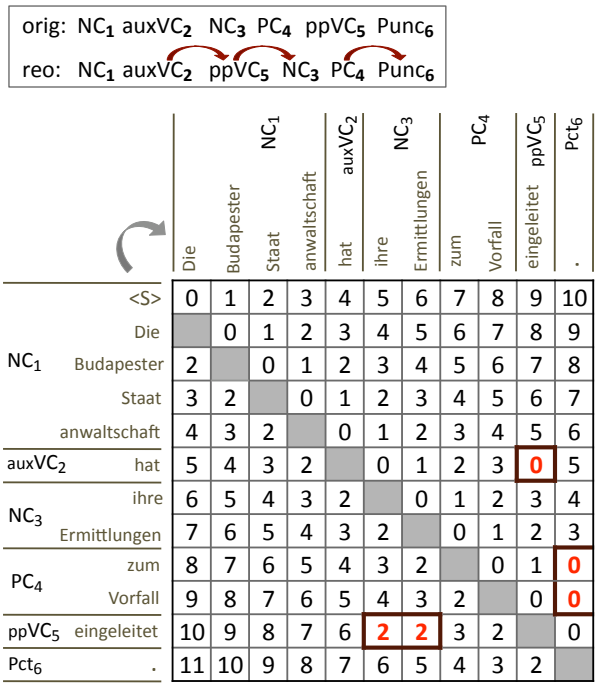


Figure 2: Modified distortion matrix (mode A×A) of the German sentence given in Fig. 1. The chunk reordering shown on top generates three *shortcuts* corresponding to the 0's and 2's highlighted in the matrix.

Fig. 2 shows the distortion matrix of the German sentence of Fig. 1, with starting positions as columns and landing positions as rows. Suppose we want to encode the reordering shown on top of Fig. 2, corresponding to the merging of the broken VC *'hat ... eingeleitet'*. This permutation contains three jumps: (2,5), (5,3) and (4,6). Converted to word-level in **A×A** mode, these yield five word shortcuts[8]: one for the onward jump (2,5) assigned 0 distortion; two for the backward jump (5,3), assigned 2; and two for the onward jump (4,6), also assigned 0. The desired reordering is now attainable within a DL of 2 words instead of 5. The same process is then applied to other permutations of the sentence.

If compared to the word reordering lattices used by Bisazza and Federico (2010) and Andreas et al. (2011), modified distortion matrices provide a more compact, implicit way to encode likely reorderings in a sentence-specific fashion. Matrix representation does not require multiplication of nodes for the same source word and is naturally compatible with the PSMT decoder's standard reordering mechanisms.

7 Evaluation

In this section we evaluate the impact of modified distortion matrices on two news translation tasks.

Matrices were integrated into the Moses toolkit (Koehn et al., 2007) using a sentence-level XML markup. The list of word shortcuts for each sentence is provided as an XML tag that is parsed by the decoder to modify the distortion matrix just before starting the search. As usual, the distortion matrix is queried by the distortion penalty generator and by the hypothesis expander[9].

7.1 Experimental setup

For **Arabic-English**, we use the union of all in-domain parallel corpora provided for the NIST-MT09 evaluation[10] for a total of 986K sentences, 31M English words. The target LM is trained on the English side of all available NIST-MT09 parallel data, UN included (147M words). For development and test, we use the newswire sections of the NIST benchmarks, hereby called dev06-NW, eval08-NW and eval09-NW: 1033, 813 and 586 sentences, respectively, each provided with four reference translations.

The **German-English** system is instead trained on WMT10 data: namely Europarl (v.5) plus News-commentary-2010 for a total of 1.6M parallel sentences, 43M English words. The target LM is trained on the monolingual news data provided for the constrained track (1133M words). For development and test, we use the WMT10 news benchmarks test08, test09 and test10: 2051, 2525 and 2489 sentences, respectively, with one reference translation.

Concerning pre-processing, we apply standard tokenization to the English data, while for Arabic we use our in-house tokenizer that removes diacritics and normalizes special characters. Arabic text is then segmented with AMIRA (Diab et al., 2004) according to the ATB scheme[11]. German tokenization

[8] In **L×F** mode, instead, each chunk-to-chunk jump would yield exactly one word shortcut, for a total of three.

[9] Note that lexicalized reordering models use real word distances to compute the orientation class of a new hypothesis, thus they are not affected by changes in the matrix.

[10] That is everything except the small GALE corpus and the UN corpus. As reported by Green et al. (2010) the removal of UN data does not affect baseline performances on news test.

[11] The Arabic Treebank tokenization scheme isolates con-

and compound splitting is performed with Tree Tagger (Schmid, 1994) and the Gertwol morphological analyser (Koskenniemi and Haapalainen, 1994)[12].

Using Moses we build competitive baselines on the training data described above. Word alignment is produced by the Berkeley Aligner (Liang et al., 2006). The decoder is based on the log-linear combination of a phrase translation model, a lexicalized reordering model, a 6-gram target language model, distortion cost, word and phrase penalties. The 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. We choose the hierarchical variant, as it was shown by its authors to outperform the default word-based on an Arabic-English task. Finally, for German, we enable the Moses option *monotone-at-punctuation* which forbids reordering across punctuation marks. The DL is initially set to 5 words for Arabic-English and to 10 for German-English. According to our experience, these are the optimal settings for the evaluated tasks. Feature weights are optimized by minimum error training (Och, 2003) on the development sets (dev06-NW and test08).

7.2 Translation quality and efficiency results

We evaluate translations with BLEU (Papineni et al., 2002) and METEOR (Banerjee and Lavie, 2005). As these scores are only indirectly sensitive to word order, we also compute KRS or Kendall Reordering Score (Birch et al., 2010; Bisazza et al., 2011) which is a positive score based on the Kendall's Tau distance between the source-output and source-reference permutations. To isolate the impact of our techniques on problematic reordering, we extract from each test set the sentences that got permuted by "oracle reordering" (see Sect. 5). These constitute about a half of the Arabic sentences, and about a third of the German. We refer to the KRS computed on these test *subsets* as KRS(R). Statistically significant differences are assessed by approximate randomization as in (Riezler and Maxwell, 2005)[13].

Tab. 1 reports results obtained by varying the DL and modifying the distortion function. To evaluate the reordering selection technique, we also compare the encoding of all rule-generated reorderings against only the 3 best per rule-matching sequence, as ranked by our best performing reordered LM (see end of Sect. 5). We mark the DL with a '+' to denote that some longer jumps are being allowed by modified distortion. Run times refer to the translation of the first 100 sentences of eval08-NW and test09 by a 4-core processor.

Arabic-English. As anticipated, raising the DL does not improve, but rather worsen performances. The decrease in BLEU and METEOR reported with DL=8 is not significant, but the decrease in KRS is both significant and large. Efficiency is heavily affected, with a 42% increase of the run time.

Results in the row "allReo" are obtained by encoding all the rule-generated reorderings in **L**×**F** chunk-to-word conversion mode. Except for some gains in KRS reported on eval08-NW, most of the scores are lower or equal to the baseline. Such inconsistent behaviour is probably due to the low precision of the Arabic rule set, pointed out in Sect. 4.

Finally, we arrive to the performance of 3-best reorderings per sequence. With **L**×**F** we obtain several improvements, but it's with **A**×**A** that we are able to beat the baseline according to all metrics. BLEU and METEOR improvements are rather small but significant and consistent across test sets, the best gain being reported on eval09-NW (+.9 BLEU). Most importantly, substantial word order improvements are achieved on both full test sets (+.7/+.6 KRS) and selected subsets (+.7/+.6 KRS(R)). According to these figures, word order is affected only in the sentences that contain problematic reordering. This is good evidence, suggesting that the decoder does not get "confused" by spurious shortcuts.

Looking at run times, we can say that modified distortion matrices are a very efficient way to address long reordering. Even when all the generated reorderings are encoded, translation time increases only by 4%. Reordering selection naturally helps to further reduce decoding overload. As for conversion modes, **A**×**A** yields slightly higher run times than **L**×**F** because it generates more shortcuts.

German-English. In this task we manage to improve translation quality with a setting that is almost

junctions *w+* and *f+*, prepositions *l+, k+, b+*, future marker *s+*, pronominal suffixes, but not the article *Al+*.

[12] http://www2.lingsoft.fi/cgi-bin/gertwol

[13] Translation scores and significance tests are computed with the tools multeval (Clark et al., 2011) and sigf (Padó, 2006).

(a) Arabic to English

Distortion Function	DL	eval08-nw bleu	met	krs	krs(R)	eval09-nw bleu	met	krs	krs(R)	runtime (s)
† plain *[baseline]*	5	44.5	34.9	81.6	82.9	49.9	38.0	84.1	84.4	1038
plain	8	44.2°	34.8	80.7$^\bullet$	82.2$^\bullet$	49.8	37.9	83.3$^\bullet$	83.5$^\bullet$	1470
† modified: allReo, L×F	5+	44.4	34.9	82.2$^\bullet$	83.7$^\bullet$	49.9	37.8$^\bullet$	84.3	84.4	1078
modified: 3bestReo, L×F	5+	44.5	35.1$^\bullet$	82.3$^\bullet$	83.5$^\bullet$	50.7$^\bullet$	38.1	84.8$^\bullet$	85.0$^\bullet$	1052
† modified: 3bestReo, A×A	5+	44.8°	35.1$^\bullet$	82.3$^\bullet$	83.6$^\bullet$	50.8$^\bullet$	38.2$^\bullet$	84.7$^\bullet$	85.0$^\bullet$	1072

(b) German to English

Distortion Function	DL	test09 bleu	met	krs	krs(R)	test10 bleu	met	krs	krs(R)	runtime (s)
† plain *[baseline]*	10	18.8	27.5	65.8	66.7	20.1	29.4	68.7	68.9	629
plain	20	18.4$^\bullet$	27.4$^\bullet$	63.6$^\bullet$	65.2 $^\bullet$	19.8$^\bullet$	29.3$^\bullet$	66.3$^\bullet$	66.6$^\bullet$	792
plain	4	18.4$^\bullet$	27.4$^\bullet$	67.3$^\bullet$	66.9	19.6$^\bullet$	29.1$^\bullet$	70.2$^\bullet$	69.6$^\bullet$	345
† modified: allReo, L×F	4+	19.1$^\bullet$	27.6$^\bullet$	67.6$^\bullet$	68.1$^\bullet$	20.4$^\bullet$	29.4	70.6$^\bullet$	70.7$^\bullet$	352
modified: 3bestReo, L×F	4+	19.2$^\bullet$	27.7$^\bullet$	67.4$^\bullet$	68.1$^\bullet$	20.4$^\bullet$	29.4	70.4$^\bullet$	70.6$^\bullet$	351
† modified: 3bestReo, A×A	4+	19.2$^\bullet$	27.7$^\bullet$	67.4$^\bullet$	68.4$^\bullet$	20.6$^\bullet$	29.5°	70.4$^\bullet$	70.7$^\bullet$	357

Table 1: Impact of modified distortion matrices on translation quality, measured with BLEU, METEOR and KRS (all in percentage form, higher scores mean higher quality). The settings used for weight tuning are marked with †. Statistically significant differences wrt the baseline are marked with $^\bullet$ at the $p \leq .05$ level and $^\circ$ at the $p \leq .10$ level.

twice as fast as the baseline. As shown by the first part of the table, the best baseline results are obtained with a rather high DL, that is 10 (only KRS improves with a lower DL). However, with modified distortion, the best results according to all metrics are obtained with a DL of 4.

Looking at the rest of the table, we see that reordering selection is not as crucial as in Arabic-English. This is in line with the properties of the more precise German reordering rule set (two rules out of three generate at most 3 reorderings per sequence). Considering all scores, the last setting (3-best reordering and **A**×**A**) appears as the best one, achieving the following gains over the baseline: +.4/+.5 BLEU, +.2/+.1 METEOR, +1.6/+1.7 KRS and +1.7/+1.8 KRS(R). The agreement observed among such diverse metrics makes us confident about the goodness of the approach.

8 Conclusions

In Arabic-English and German-English, long reordering concentrates on specific patterns describable by a small number of linguistic rules. By means of non-deterministic chunk reordering rules, we have generated likely permutations of the test sentences and ranked them with n-gram LMs trained on pre-reordered data. We have then introduced the notion of modified distortion matrices to naturally encode a set of likely reorderings in the decoder input. Modified distortion allows for a finer and more linguistically informed definition of the search space, which is reflected in better translation outputs and more efficient decoding.

We expect that further improvements may be achieved by refining the Arabic reordering rules with specific POS tags and lexical cues. We also plan to evaluate modified distortion matrices in conjunction with a different type of in-decoding reordering model such as the one proposed by Green et al. (2010). Finally, we may try to exploit not only the ranking, but also the scores produced by the reordered LMs, as an additional decoding feature.

Acknowledgments

This work was supported by the T4ME network of excellence (IST-249119) funded by the European Commission DG INFSO through the 7^{th} Framework Programme. We thank Christian Hardmeier for helping us define the German reordering rules, and the anonymous reviewers for valuable suggestions.

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. Association for Computational Linguistics.

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. Association for Computational Linguistics.

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. Association for Computational Linguistics.

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. 2010. Chunk-based verb reordering in VSO sentences for Arabic-English statistical machine translation. In *Proceedings of the Joint Fifth Workshop on Statistical Machine Translation and Metrics MATR*, pages 241–249, Uppsala, Sweden, July. Association for Computational Linguistics.

Arianna Bisazza, Daniele Pighin, and Marcello Federico. 2011. Chunk-lattices for verb reordering in Arabic-English statistical machine translation. *Machine Translation*, Published Online.

David Chiang. 2005. A hierarchical phrase-based model for statistical machine translation. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05)*, pages 263–270, Ann Arbor, Michigan, June. Association for Computational Linguistics.

Jonathan Clark, Chris Dyer, Alon Lavie, and Noah Smith. 2011. Better hypothesis testing for statistical machine translation: Controlling for optimizer instability. In *Proceedings of the Association for Computational Lingustics*, ACL 2011, Portland, Oregon, USA. Association for Computational Linguistics. accepted; available at http://www.cs.cmu.edu/ jhclark/pubs/significance.pdf.

Michael Collins, Philipp Koehn, and Ivona Kucerova. 2005. Clause restructuring for statistical machine translation. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05)*, pages 531–540, Ann Arbor, Michigan, June. Association for Computational Linguistics.

Josep M. Crego and Nizar Habash. 2008. Using shallow syntax information to improve word alignment and reordering for smt. In *StatMT '08: Proceedings of the Third Workshop on Statistical Machine Translation*, pages 53–61, Morristown, NJ, USA. Association for Computational Linguistics.

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, May 2 - May 7. Association for Computational Linguistics.

Jakob Elming and Nizar Habash. 2009. Syntactic reordering for English-Arabic phrase-based machine translation. In *Proceedings of the EACL 2009 Workshop on Computational Approaches to Semitic Languages*, pages 69–77, Athens, Greece, March. Association for Computational Linguistics.

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. Association for Computational Linguistics.

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. Association for Computational Linguistics.

Nizar Habash. 2007. Syntactic preprocessing for statistical machine translation. In Bente Maegaard, editor, *Proceedings of the Machine Translation Summit XI*, pages 215–222, Copenhagen, Denmark.

Christian Hardmeier, Arianna Bisazza, and Marcello Federico. 2010. FBK at WMT 2010: Word lattices for morphological reduction and chunk-based reordering. In *Proceedings of the Joint Fifth Workshop on Statistical Machine Translation and Metrics MATR*, pages 88–92, Uppsala, Sweden, July. Association for Computational Linguistics.

Philipp Koehn, Franz Josef Och, and Daniel Marcu. 2003. Statistical phrase-based translation. In *Proceed-*

ings 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.

Kimmo Koskenniemi and Mariikka Haapalainen, 1994. *GERTWOL – Lingsoft Oy*, chapter 11, pages 121–140. Roland Hausser, Niemeyer, Tübingen.

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. Association for Computational Linguistics.

Jan Niehues and Muntsin Kolss. 2009. A POS-based model for long-range reorderings in SMT. In *Proceedings of the Fourth Workshop on Statistical Machine Translation*, pages 206–214, Athens, Greece, March. Association for Computational Linguistics.

Franz Josef Och. 2002. *Statistical Machine Translation: From Single-Word Models to Alignment Templates*. Ph.D. thesis, RWTH Aachen, Germany.

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.

Sebastian Padó, 2006. *User's guide to `sigf`: Significance testing by approximate randomisation*.

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. Association for Computational Linguistics.

Helmut Schmid. 1994. Probabilistic part-of-speech tagging using decision trees. In *Proceedings of International Conference on New Methods in Language Processing*.

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)*.

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

Fei Xia and Michael McCord. 2004. Improving a statistical MT system with automatically learned rewrite patterns. In *Proceedings of Coling 2004*, pages 508–514, Geneva, Switzerland, Aug 23–Aug 27. COLING.

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.

Kenji Yamada. 2002. *A syntax-based translation model*. Ph.D. thesis, Department of Computer Science, University of Southern California, Los Angeles.

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. Association for Computational Linguistics.

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.

Yuqi Zhang, Richard Zens, and Hermann Ney. 2007. Chunk-level reordering of source language sentences with automatically learned rules for statistical machine translation. In *Proceedings of SSST, NAACL-HLT 2007 / AMTA Workshop on Syntax and Structure in Statistical Translation*, pages 1–8, Rochester, New York, April. Association for Computational Linguistics.

Andreas Zollmann, Ashish Venugopal, Franz Och, and Jay Ponte. 2008. A systematic comparison of phrase-based, hierarchical and syntax-augmented statistical MT. In *Proceedings of the 22nd International Conference on Computational Linguistics (Coling 2008)*, pages 1145–1152, Manchester, UK, August. Coling 2008 Organizing Committee.