

# A Discriminative Syntactic Model for Source Permutation via Tree Transduction

Maxim Khalilov and Khalil Sima'an

Institute for Logic, Language and Computation

University of Amsterdam

P.O. Box 94242

1090 GE Amsterdam, The Netherlands

{m.khalilov, k.simaan}@uva.nl

## Abstract

A major challenge in statistical machine translation is mitigating the word order differences between source and target strings. While reordering and lexical translation choices are often conducted in tandem, *source string permutation* prior to translation is attractive for studying reordering using hierarchical and syntactic structure. This work contributes an approach for learning source string permutation via transfer of the source syntax tree. We present a novel discriminative, probabilistic tree transduction model, and contribute a set of empirical upperbounds on translation performance for English-to-Dutch source string permutation under sequence and parse tree constraints. Finally, the translation performance of our learning model is shown to outperform the state-of-the-art phrase-based system significantly.

## 1 Introduction

From its beginnings, statistical machine translation (SMT) has faced a word reordering challenge that has a major impact on translation quality. While standard mechanisms embedded in phrase-based SMT systems, e.g. (Och and Ney, 2004), deal efficiently with word reordering within a limited window of words, they are still not expected to handle all possible reorderings that involve words beyond this relatively narrow window, e.g., (Tillmann and Ney, 2003; Zens and Ney, 2003; Tillman, 2004). More recent work handles word

order differences between source and target languages using hierarchical methods that draw on Inversion Transduction Grammar (ITG), e.g., (Wu and Wong, 1998; Chiang, 2005). In principle, the latter approach explores reordering defined by the choice of swapping the order of sibling subtrees under each node in a binary parse-tree of the source/target sentence.

An alternative approach aims at minimizing the need for reordering during translation by permuting the source sentence as a pre-translation step, e.g., (Collins et al., 2005; Xia and McCord, 2004; Wang et al., 2007; Khalilov, 2009). In effect, the translation process works with a model for source permutation ( $s \rightarrow s'$ ) followed by translation model ( $s' \rightarrow t$ ), where  $s$  and  $t$  are source and target strings and  $s'$  is the target-like permuted source string. In how far can source permutation reduce the need for reordering in conjunction with translation is an empirical question.

In this paper we define source permutation as the problem of learning how to *transfer* a given source parse-tree into a parse-tree that minimizes the divergence from target word-order. We model the tree transfer  $\tau_s \rightarrow \tau_{s'}$  as a sequence of local, independent transduction operations, each transforming the current intermediate tree  $\tau_{s'_i}$  into the next intermediate tree  $\tau_{s'_{i+1}}$ , with  $\tau_{s_0} = \tau_s$  and  $\tau_{s'_n} = \tau_{s'}$ . A transduction operation merely permutes the sequence of  $n > 1$  children of a single node in an intermediate tree, i.e., unlike previous work, we do not binarize the trees. The number of permutations is factorial in  $n$ , and learning a sequence of transductions for explaining a source permutation can be computationally rather challenging (see (Tromble and Eisner, 2009)). Yet,

from the limited perspective of source *string* permutation ( $s \rightarrow s'$ ), another challenge is to integrate a figure of merit that measures in how far  $s'$  resembles a plausible target word-order.

We contribute solutions to these challenging problems. Firstly, we learn the transduction operations using a discriminative estimate of  $P(\pi(\alpha_x) | N_x, \alpha_x, context_x)$ , where  $N_x$  is the label of node (address)  $x$ ,  $N_x \rightarrow \alpha_x$  is the context-free production under  $x$ ,  $\pi(\alpha_x)$  is a permutation of  $\alpha_x$  and  $context_x$  represents a surrounding syntactic context. As a result, this constrains  $\{\pi(\alpha_x)\}$  only to those found in the training data, and it conditions the transduction application probability on its specific contexts. Secondly, in every sequence  $s'_0 = s, \dots, s'_n = s'$  resulting from a tree transductions, we prefer those local transductions on  $\tau_{s'_{i-1}}$  that lead to source string permutation  $s'_i$  that are closer to target word order than  $s'_{i-1}$ ; we employ  $s'$  language model probability ratios as a measure of word order improvement.

In how far does the assumption of source permutation provide any window for improvement over a phrase-based translation system? We conduct experiments on translating from English into Dutch, two languages which are characterized by a number of systematic divergences between them. Initially, we conduct oracle experiments with varying constraints on source permutation to set upperbounds on performance relative to a state-of-the-art system. Translating the oracle source string permutation (obtained by untangling the crossing alignments) offers a large margin of improvement, whereas the oracle parse tree permutation provides a far smaller improvement. A minor change to the latter to also permute constituents that include words aligned with NULL, offers further improvement, yet lags behind bare string permutation. Subsequently, we present translation results using our learning approach, and exhibit a significant improvement in BLEU score over the state-of-the-art baseline system. Our analysis shows that syntactic structure can provide important clues for reordering in translation, especially for dealing with long distance cases found in, e.g., English and Dutch. Yet, tree transduction by merely permuting the order of sis-

ter subtrees might turn out insufficient.

## 2 Baseline: Phrase-based SMT

Given a word-aligned parallel corpus, phrase-based systems (Och and Ney, 2002; Koehn et al., 2003) work with (in principle) arbitrarily large phrase pairs (also called blocks) acquired from word-aligned parallel data under a simple definition of translational equivalence (Zens et al., 2002). The conditional probabilities of one phrase given its counterpart are interpolated log-linearly together with a set of other model estimates:

$$\hat{e}_1^I = \arg \max_{e_1^I} \left\{ \sum_{m=1}^M \lambda_m h_m(e_1^I, f_1^J) \right\} \quad (1)$$

where a feature function  $h_m$  refer to a system model, and the corresponding  $\lambda_m$  refers to the relative weight given to this model. A phrase-based system employs feature functions for a phrase pair translation model, a language model, a reordering model, and a model to score translation hypothesis according to length. The weights  $\lambda_m$  are usually optimized for system performance (Och, 2003) as measured by BLEU (Papineni et al., 2002). Two reordering methods are widely used in phrase-based systems.

**Distance-based** A simple distance-based reordering model default for Moses system is the first reordering technique under consideration. This model provides the decoder with a cost linear to the distance between words that should be reordered.

**MSD** A lexicalized block-oriented data-driven reordering model (Tillman, 2004) considers three different orientations: monotone (M), swap (S), and discontinuous (D). The reordering probabilities are conditioned on the lexical context of each phrase pair, and decoding works with a block sequence generation process with the possibility of swapping a pair of blocks.

## 3 Related Work on Source Permutation

The integration of linguistic syntax into SMT systems offers a potential solution to reordering problem. For example, syntax is successfully integrated into hierarchical SMT (Zollmann and

Venugopal, 2006). Similarly, the tree-to-string syntax-based transduction approach offers a complete translation framework (Galley et al., 2006).

The idea of augmenting SMT by a reordering step prior to translation has often been shown to improve translation quality. Clause restructuring performed with hand-crafted reordering rules for German-to-English and Chinese-to-English tasks are presented in (Collins et al., 2005) and (Wang et al., 2007), respectively. In (Xia and McCord, 2004; Khalilov, 2009) word reordering is addressed by exploiting syntactic representations of source and target texts.

Other reordering models operate provide the decoder with multiple word orders. For example, the MaxEnt reordering model described in (Xiong et al., 2006) provides a hierarchical phrasal reordering system integrated within a CKY-style decoder. In (Galley and Manning, 2008) the authors present an extension of the famous MSD model (Tillman, 2004) able to handle long-distance word-block permutations. Coming up-to-date, in (PVS, 2010) an effective application of data mining techniques to syntax-driven source reordering for MT is presented.

Recently, Tromble and Eisner (2009) define source permutation as learning source permutations; the model works with a preference matrix for word pairs, expressing preference for their two alternative orders, and a corresponding weight matrix that is fit to the parallel data. The huge space of permutations is then structured using a binary synchronous context-free grammar (Binary ITG) with  $O(n^3)$  parsing complexity, and the permutation score is calculated recursively over the tree at every node as the accumulation of the relative differences between the word-pair scores taken from the preference matrix. Application to German-to-English translation exhibits some performance improvement.

Our work is in the general learning direction taken in (Tromble and Eisner, 2009) but differs both in defining the space of permutations, using local probabilistic tree transductions, as well as in the learning objective aiming at scoring permutations based on a log-linear interpolation of a local syntax-based model with a global string-based (language) model.

## 4 Pre-Translation Source Permutation

Given a word-aligned parallel corpus, we define the source string permutation as the task of learning to unfold the crossing alignments between sentence pairs in the parallel corpus. Let be given a source-target sentence pair  $s \rightarrow t$  with word alignment set  $a$  between their words. Unfolding the crossing instances in  $a$  should lead to as monotone an alignment  $a'$  as possible between a permutation  $s'$  of  $s$  and the target string  $t$ . Conducting such a “monotonization” on the parallel corpus gives two parallel corpora: (1) a source-to-permutation parallel corpus ( $s \rightarrow s'$ ) and (2) a source permutation-to-target parallel corpus ( $s' \rightarrow t$ ). The latter corpus is word-aligned automatically again and used for training a phrase-based translation system, while the former corpus is used for training our model for pre-translation source permutation via parse tree transductions.

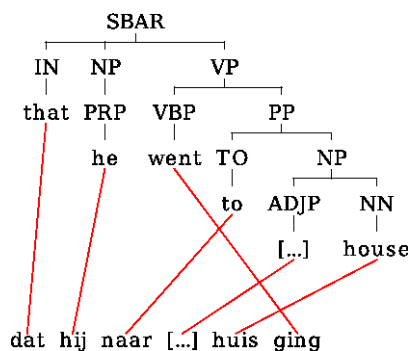


Figure 1: Example of crossing alignments and long-distance reordering using a source parse tree.

In itself, the problem of permuting the source string to unfold the crossing alignments is computationally intractable (see (Tromble and Eisner, 2009)). However, different kinds of constraints can be made on unfolding the crossing alignments in  $a$ . A common approach in hierarchical SMT is to assume that the source string has a binary parse tree, and the set of eligible permutations is defined by binary ITG transductions on this tree. This defines permutations that can be obtained only by at most inverting pairs of children under nodes of the source tree. Figure 1 exhibits a long distance reordering of the verb in English-to-Dutch translation: inverting the order of the children under the VP node would unfold the crossing alignment.

#### 4.1 Oracle Performance

As has been shown in the literature (Costa-jussà and Fonollosa, 2006; Khalilov and Sima’an, 2010; Wang et al., 2007), source and target texts monotonization leads to a significant improvement in terms of translation quality. However it is not known how many alignment crossings can be unfolded under different parse tree conditions. In order to gauge the impact of corpus monotonization on translation system performance, we trained a set of oracle translation systems, which create target sentences that follow the source language word order using the word alignment links and various constraints.

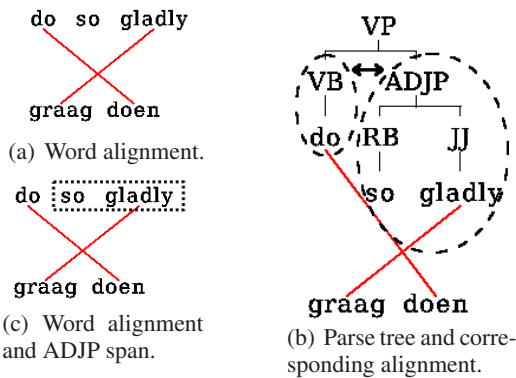


Figure 2: Reordering example.

The set-up of our experiments and corpus characteristics are detailed in Section 5. Table 1 reports translation scores of the oracle systems. Notice that all the numbers are calculated on the realigned corpora. Baseline results are provided for informative purposes.

**String permutation** The first oracle system under consideration is created by traversing the string from left to right and unfolding all crossing alignment links (we call this system *oracle-string*). For example in Figure 2(a), the *oracle-string* system generates a string “do so gladly” swapping the words “do” and “gladly” without considering the parse tree. The first line of the table shows the performance of the *oracle-string* system with monotone source and target portions of the corpus.

**Oracle under tree constraint** We use a syntactic parser for parsing the English source sentences

that provide  $n$ -ary constituency parses. Now we constrain unfolding crossing alignments only to those alignment links which agree with the structure of the source-side parse tree and consider the constituents which include aligned tokens only. Unfolding a crossing alignment is modeled as permuting the children of a node in the parse tree. We refer to this oracle system as *oracle-tree*. For example provided in Figure 2(b), there is no way to construct a monotonized version of the sentence since the word “so” is aligned to NULL and impedes swapping the order of VB and ADJP under the VP.

**Oracle under relaxed tree constraint** The *oracle-tree* system does not permute the words which are both (1) not found in the alignment and (2) are spanned by the sub-trees sibling to the reordering constituents. Now we introduce a relaxed version of the parse tree constraint: the order of the children of a node is permuted when the node covers the reordering constituents and *also* when the frontier contains leaf nodes aligned with NULL (*oracle-span*). For example, in Figure 2(c) the English word “so” is not aligned, but according to the relaxed version, must move together with the word “gladly” since they share a parent node (*ADJP*).

Source	BLEU	NIST
<i>baseline dist</i>	24.04	6.29
<i>baseline MSD</i>	24.04	6.28
<i>oracle - string</i>	27.02	6.51
<i>oracle - tree</i>	24.09	6.30
<i>oracle - span</i>	24.95	6.37

Table 1: Translation scores of oracle systems.

The main conclusion which can be drawn from the oracle results is that there is a possibility for relatively big ( $\approx 3$  BLEU points) improvement with complete unfolding of crossing alignments and very limited ( $\approx 0.05$  BLEU points) with the same done under the parse tree constraint. A tree-based system that allows for permuting unaligned words that are covered by a dominating parent node shows more improvement in terms of BLEU and NIST scores ( $\approx 0.9$  BLEU points).

The gap between *oracle-string* and *oracle-tree* performance is due to alignment crossings which

cannot be unfolded under trees (illustrated in Figure 3), but possibly also due to parse and alignment errors.

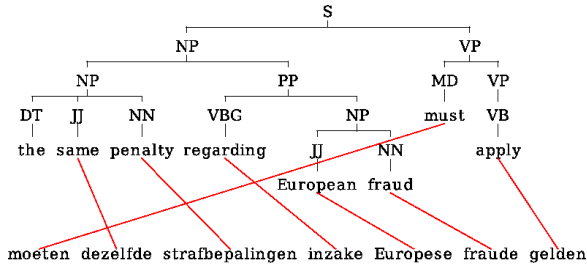


Figure 3: Example of alignment crossing that does not agree with the parse tree.

## 4.2 Source Permutation via Syntactic Transfer

Given a parallel corpus with string pairs  $s \rightarrow t$  with word alignment  $a$ , we create a *source permuted* parallel corpus  $s \rightarrow s'$  by unfolding the crossing alignments in  $a$ : this is done by scanning the string  $s$  from left to right and moving words involved in crossing alignments to positions where the crossing alignments are unfolded). The source strings  $s$  are parsed, leading to a single parse tree  $\tau_s$  per source string.

Our model aims at learning from the source permuted parallel corpus  $s \rightarrow s'$  a probabilistic optimization  $\arg \max_{\pi(s)} P(\pi(s) | s, \tau_s)$ . We assume that the set of permutations  $\{\pi(s)\}$  is defined through a finite set of local transductions over the tree  $\tau_s$ . Hence, we view the permutations leading from  $s$  to  $s'$  as a sequence of local tree transductions  $\tau_{s'_0} \rightarrow \dots \rightarrow \tau_{s'_n}$ , where  $s'_0 = s$  and  $s'_n = s'$ , and each transduction  $\tau_{s'_{i-1}} \rightarrow \tau_{s'_i}$  is defined using a tree transduction operation that *at most permutes the children of a single node in  $\tau_{s'_{i-1}}$  as defined next.*

A local transduction  $\tau_{s'_{i-1}} \rightarrow \tau_{s'_i}$  is modelled by an operation that applies to a single node with address  $x$  in  $\tau_{s'_{i-1}}$ , labeled  $N_x$ , and may permute the ordered sequence of children  $\alpha_x$  dominated by node  $x$ . This constitutes a direct generalization of the ITG binary inversion transduction operation. We assign a conditional probability to each such

local transduction:

$$P(\tau_{s'_i} | \tau_{s'_{i-1}}) \approx P(\pi(\alpha_x) | N_x \rightarrow \alpha_x, C_x) \quad (2)$$

where  $\pi(\alpha_x)$  is a permutation of  $\alpha_x$  (the ordered sequence of node labels under  $x$ ) and  $C_x$  is a local tree context of node  $x$  in tree  $\tau_{s'_{i-1}}$ . One wrinkle in this definition is that the number of possible permutations of  $\alpha_x$  is factorial in the length of  $\alpha_x$ . Fortunately, the source permuted training data exhibits only a fraction of possible permutations even for longer  $\alpha_x$  sequences. Furthermore, by conditioning the probability on local context, the general applicability of the permutation is restrained.

Given this definition, we define the probability of the sequence of local tree transductions  $\tau_{s'_0} \rightarrow \dots \rightarrow \tau_{s'_n}$  as

$$P(\tau_{s'_0} \rightarrow \dots \rightarrow \tau_{s'_n}) = \prod_{i=1}^n P(\tau_{s'_i} | \tau_{s'_{i-1}}) \quad (3)$$

The problem of calculating the most likely permutation under this transduction model is made difficult by the fact that different transduction sequences may lead to the same permutation, which demands summing over these sequences. Furthermore, because every local transduction conditions on local context of an intermediate tree, this quickly risks becoming intractable (even when we use packed forests). In practice we take a pragmatic approach and greedily select at every intermediate point  $\tau_{s'_{i-1}} \rightarrow \tau_{s'_i}$  the single most likely local transduction that can be conducted on any node of the current intermediate tree  $\tau_{s'_{i-1}}$  using an interpolation of the term in Equation 2 with string probability ratios as follows:

$$P(\pi(\alpha_x) | N_x \rightarrow \alpha_x, C_x) \times \frac{P(s'_{i-1})}{P(s'_i)}$$

The rationale behind this log-linear interpolation is that our source permutation approach aims at finding the optimal permutation  $s'$  of  $s$  that can serve as input for a subsequent translation model. Hence, we aim at tree transductions that are syntactically motivated that also lead to improved string permutation. In this sense, the tree transduction definitions can be seen as an efficient and

syntactically informed way to define the space of possible permutations.

We estimate the string probabilities  $P(s'_i)$  using 5-gram language models trained on the  $s'$  side of the source permuted parallel corpus  $s \rightarrow s'$ . We estimate the conditional probability  $P(\pi(\alpha_x) \mid N_x \rightarrow \alpha_x, C_x)$  using a Maximum-Entropy framework, where feature functions are defined to capture the permutation as a class, the node label  $N_x$  and its head POS tag, the child sequence  $\alpha_x$  together with the corresponding sequence of head POS tags and other features corresponding to different contextual information.

We were particularly interested in those linguistic features that motivate reordering phenomena from the syntactic and linguistic perspective. The features that were used for training the permutation system are extracted for every internal node of the source tree that has more than one child:

- *Local tree topology.* Sub-tree instances that include parent node and the ordered sequence of child node labels.
- *Dependency features.* Features that determine the POS tag of the head word of the current node, together with the sequence of POS tags of the head words of its child nodes.
- *Syntactic features.* Three binary features from this class describe: (1) whether the parent node is a child of the node annotated with the same syntactic category, (2) whether the parent node is a descendant of the node annotated with the same syntactic category, and (3) if the current subtree is embedded into a “*SENT-SBAR*” sub-tree. The latter feature intends to model the divergence in word order in relative clauses between Dutch and English which is illustrated in Figure 1.

In initial experiments we piled up all feature functions into a single model. Preliminary results showed that the system performance increases if the set of patterns is split into partial classes conditioned on the current node label. Hence, we trained four separate MaxEnt models for the categories with potentially high number of crossing alignments, namely *VP*, *NP*, *SENT*, and *SBAR*.

For combinatory models we use the following notations:  $M_4 = \sum_{i \in \{NP, VP, SENT, SBAR\}} M_i$  and  $M_2 = \sum_{i \in \{VP, SENT\}} M_i$ .

## 5 Experiments and results

The SMT system used in the experiments was implemented within the open-source MOSES toolkit (Koehn et al., 2007). Standard training and weight tuning procedures which were used to build our system are explained in details on the MOSES web page<sup>1</sup>. The MSD model was used together with a distance-based reordering model. Word alignment was estimated with GIZA++ tool<sup>2</sup> (Och, 2003), coupled with mkcls<sup>3</sup> (Och, 1999), which allows for statistical word clustering for better generalization. An 5-gram target language model was estimated using the SRI LM toolkit (Stolcke, 2002) and smoothed with modified Kneser-Ney discounting. We use the Stanford parser<sup>4</sup> (Klein and Manning, 2003) as a source-side parsing engine. The parser was trained on the English treebank set provided with 14 syntactic categories and 48 POS tags. The evaluation conditions were case-sensitive and included punctuation marks. For Maximum Entropy modeling we used the maxent toolkit<sup>5</sup>.

**Data** The experiment results were obtained using the English-Dutch corpus of the European Parliament Plenary Session transcription (*EuroParl*). Training corpus statistics can be found in Table 2.

	Dutch	English
Sentences	1.2 M	1.2 M
Words	32.9 M	33.0 M
Average sentence length	27.20	27.28
Vocabulary	228 K	104 K

Table 2: Basic statistics of the English-Dutch EuroParl training corpus.

The development and test datasets were randomly chosen from the corpus and consisted of

<sup>1</sup><http://www.statmt.org/moses/>

<sup>2</sup>[code.google.com/p/giza-pp/](http://code.google.com/p/giza-pp/)

<sup>3</sup><http://www.fjoch.com/mkcls.html>

<sup>4</sup><http://nlp.stanford.edu/software/lex-parser.shtml>

<sup>5</sup>[http://homepages.inf.ed.ac.uk/lzhang10/maxent\\_toolkit.html](http://homepages.inf.ed.ac.uk/lzhang10/maxent_toolkit.html)

500 and 1,000 sentences, respectively. Both were provided with one reference translation.

**Results** Evaluation of the system performance is twofold. In the first step, we analyze the quality of reordering method itself. In the next step we look at the automatic translation scores and evaluate the impact which the choice of reordering strategy has on the translation quality. In both stages of evaluation, the results are contrasted with the performance shown by the standard phrase-based SMT system (*baseline*) and with oracle results.

**Source reordering analysis** Table 3 shows the parameters of the reordered system allowing to assess the effectiveness of reordering permutations, namely: (1) a total number of crossings found in the word alignment (#C), (2) the size of the resulting phrase table (PT), (3) BLEU, NIST, and WER scores obtained using monotonized parallel corpus (*oracle*) as a reference.

All the numbers are calculated on the re-aligned corpora. Calculations are done on the basis of the 100,000 line extraction from the corpus<sup>6</sup> and corresponding alignment matrix. The *baseline* rows show the number of alignment crossings found in the original (unmonotonized) corpus.

System	#C	PT	Scores		
			BLEU	NIST	WER
Oracle					
string	54.6K	48.4M	-	-	-
tree	187.3K	30.3M	71.73	17.01	16.77
span	146.9K	33.0M	73.41	17.11	15.73
Baselines					
baselines	187.0K	29.8M	71.70	17.07	16.55
Category models					
$M_{NP}$	188.9K	29.7M	71.63	17.07	16.52
$M_{VP}$	168.1K	29.8M	73.17	17.16	15.99
$M_{SENT}$	171.0K	29.8M	73.08	17.08	16.10
$M_{SBAR}$	188.6K	29.8M	72.89	16.90	16.41
Combinatory models					
$M_4$	193.2K	29.1M	70.98	16.85	16.78
$M_2$	165.4K	29.9M	73.07	16.92	15.88

Table 3: Main parameters of the tree-based reordering system.

<sup>6</sup>A smaller portion of the corpus is used for analysis in order to reduce evaluation time.

**Translation scores** The evaluation results for the development and test corpora are reported in Table 4. They include two *baseline* configurations (*dist* and *MSD*), *oracle* results and contrasts them with the performance shown by different combinations of single-category tree-based reordering models. Best scores within each experimental section are placed in cells filled with grey.

System	Dev	Test	
	BLEU	BLEU	NIST
baseline dist	23.88	24.04	6.29
baseline MSD	24.07	24.04	6.28
oracle-string	26.28	27.02	6.50
oracle-tree	23.84	24.09	6.30
oracle-span	24.79	24.95	6.35
$M_{NP}$	23.79	23.81	6.27
$M_{VP}$	24.16	24.55	6.29
$M_{SENT}$	24.27	24.56	6.32
$M_{SBAR}$	23.99	24.12	6.27
$M_4$	23.50	23.86	6.29
$M_2$	24.28	24.64	6.33

Table 4: Experimental results.

**Analysis** The number of crossings found in word alignment intersection and BLEU/NIST/WER scores estimated on reordered data vs. monotonized data report the reordering algorithm effectiveness. A big gap between number of crossings and total number of reorderings per corpus found in *oracle-string* system<sup>7</sup> and *baseline* systems demonstrates the possible reduction of system’s non-monotonicity. The difference in number of crossings and BLEU/NIST/WER scores between the *oracle-span* and the best performing MaxEnt models (namely,  $M_2$ ) shows the level of performance of the prediction module.

A number of distinct phrase translation pairs in the translation table implicitly reveals the generalization capabilities of the translation system since it simplifies the translation task. From the other hand, increased number of shorter phrases can add noise in the reordered data and makes decoding more complex. Hence, the size of phrase table itself can not be considered as a robust indicator of its translation potential.

<sup>7</sup>The number of crossings for *oracle* configuration is not zero since this parameter is calculated on the re-aligned corpus.

Table 4 shows that three of six MaxEnt reordering systems outperform *baseline* systems by about 0.5-0.6 BLEU points, that is statistically significant<sup>8</sup>. The combination of *NP*, *NP*, *SENT*, and *SBAR* models do not show good performance possibly due to increased sparseness of reordering patterns. However, the system that consider only the  $M_{VP}$  and  $M_{SENT}$  models achieves 0.62 BLEU score gain over the *baseline* configurations.

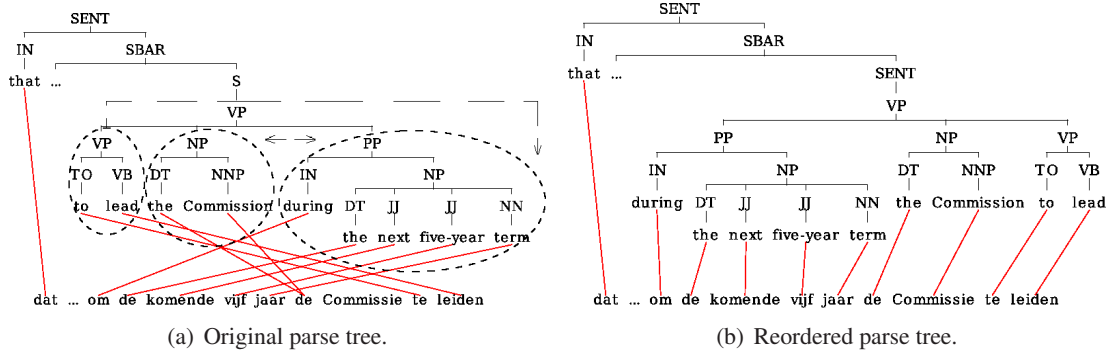
The main conclusion which can be drawn from analysis of Tables 3 and 4 is that there is an evident correlation between characteristics of reordering system and performance demonstrated by the translation system trained on the corpus with reordered source part.

**Example** Figure 4 exemplifies the sentences that presumably benefits from the monotonicization of the source part of the parallel corpus. The example demonstrates a pervading syntactic distinction between English and Dutch: the reordering of verb-phrase constituents *VP NP PP* within the relative clause into *PP NP VP*.

## 6 Conclusions and future work

We introduced a tree-based reordering model that aims at monotonicizing the word order of source

<sup>8</sup>All statistical significance calculations are done for a 95% confidence interval and 1 000 resamples, following the guidelines from (Koehn, 2004).



**Src:** *that ... to lead the Commission during the next five-year term*  
**Ref.:** *dat ... om de komende vijf jaar de Commissie te leiden*  
**Baseline MSD:** *dat ... om het voortouw te nemen in de Commissie tijdens de komende vijf jaar*  
**Rrd src:** *that ... during the next five-year term the Commission to lead*  
**M<sub>2</sub>:** *dat ... om de Commissie tijdens de komende vijf jaar te leiden*

(c) Translations.

Figure 4: Example of tree-based monotonicization.

and target languages as a pre-translation step. Our model avoids complete generalization of reordering instances by using tree contexts and limiting the permutations to data instances. From a learning perspective, our work shows that navigating a large space of intermediate tree transformations can be conducted effectively using both the source-side syntactic tree and a language model of the idealized (target-like) source-permuted language.

We have shown the potential for translation quality improvement when target sentences are created following the source language word order ( $\approx 3$  BLEU points over the standard phrase-based SMT) and under parse tree constraint ( $\approx 0.9$  BLEU points). As can be seen from these results, our model exhibits competitive translation performance scores compared with the standard distance-based and lexical reordering.

The gap between the oracle and our system's results leaves room for improvement. We intend to study extensions of the current tree transfer model to narrow this performance gap. As a first step we are combining isolated models for concrete syntactic categories and aggregating more features into the MaxEnt model. Algorithmic improvements, such as beam-search and chart parsing, could allow us to apply our method to full parse-forests as opposed to a single parse tree.

## References

- D. Chiang. 2005. A hierarchical phrase-based model for statistical machine translation. In *Proceedings of ACL'05*, pages 263–270.
- M. Collins, P. Koehn, and I. Kučerová. 2005. Clause restructuring for statistical machine translation. In *Proceedings of ACL'05*, pages 531–540.
- M. R. Costa-jussà and J. A. R. Fonollosa. 2006. Statistical machine reordering. In *Proceedings of HLT/EMNLP'06*, pages 70–76.
- M. Galley and Ch. D. Manning. 2008. A simple and effective hierarchical phrase reordering model. In *Proceedings of EMNLP'08*, pages 848–856.
- M. Galley, J. Graehl, K. Knight, D. Marcu, S. DeNeefe, W. Wang, and I. Thayer. 2006. Scalable inference and training of context-rich syntactic translation models. In *Proc. of COLING/ACL'06*, pages 961–968.
- M. Khalilov and K. Sima'an. 2010. Source reordering using maxent classifiers and supertags. In *Proc. of EAMT'10*, pages 292–299.
- M. Khalilov. 2009. *New statistical and syntactic models for machine translation*. Ph.D. thesis, Universitat Politècnica de Catalunya, October.
- D. Klein and C. Manning. 2003. Accurate unlexicalized parsing. In *Proceedings of the 41st Annual Meeting of the ACL'03*, pages 423–430.
- Ph. Koehn, F. Och, and D. Marcu. 2003. Statistical phrase-based machine translation. In *Proceedings of the HLT-NAACL 2003*, pages 48–54.
- Ph. 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 ACL 2007*, pages 177–180.
- Ph. Koehn. 2004. Statistical significance tests for machine translation evaluation. In *Proceedings of EMNLP'04*, pages 388–395.
- F. Och and H. Ney. 2002. Discriminative training and maximum entropy models for statistical machine translation. In *Proceedings of ACL'02*, pages 295–302.
- F. Och and H. Ney. 2004. The alignment template approach to statistical machine translation. *Computational Linguistics*, 30(4):417–449.
- F. Och. 1999. An efficient method for determining bilingual word classes. In *Proceedings of ACL 1999*, pages 71–76.
- F. Och. 2003. Minimum error rate training in statistical machine translation. In *Proceedings of ACL'03*, pages 160–167.
- K. Papineni, S. Roukos, T. Ward, and W. Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of ACL'02*, pages 311–318.
- A. PVS. 2010. A data mining approach to learn reorder rules for SMT. In *Proceedings of NAACL/HLT'10*, pages 52–57.
- A. Stolcke. 2002. SRILM: an extensible language modeling toolkit. In *Proceedings of SLP'02*, pages 901–904.
- C. Tillman. 2004. A unigram orientation model for statistical machine translation. In *Proceedings of HLT-NAACL'04*, pages 101–104.
- C. Tillmann and H. Ney. 2003. Word reordering and a dynamic programming beam search algorithm for statistical machine translation. *Computational Linguistics*, 1(29):93–133.
- R. Tromble and J. Eisner. 2009. Learning linear ordering problems for better translation. In *Proceedings of EMNLP'09*, pages 1007–1016.
- C. Wang, M. Collins, and Ph. Koehn. 2007. Chinese syntactic reordering for statistical machine translation. In *Proceedings of EMNLP-CoNLL'07*, pages 737–745.
- D. Wu and H. Wong. 1998. Machine translation with a stochastic grammatical channel. In *Proceedings of ACL-COLING'98*, pages 1408–1415.
- F. Xia and M. McCord. 2004. Improving a statistical MT system with automatically learned rewrite patterns. In *Proceedings of COLING'04*, pages 508–514.
- D. Xiong, Q. Liu, and S. Lin. 2006. Maximum entropy based phrase reordering model for statistical machine translation. In *Proceedings of ACL'06*, pages 521–528.
- R. Zens and H. Ney. 2003. A comparative study on reordering constraints in statistical machine translation. In *Proceedings of ACL'03*, pages 144–151.
- R. Zens, F. Och, and H. Ney. 2002. Phrase-based statistical machine translation. In *Proceedings of KI: Advances in Artificial Intelligence*, pages 18–32.
- A. Zollmann and A. Venugopal. 2006. Syntax augmented machine translation via chart parsing. In *Proceedings of NAACL'06*, pages 138–141.