Context-Sensitive Syntactic Source-Reordering by Statistical 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

How well can a phrase translation model perform if we permute the source words to fit target word order as perfectly as word alignment might allow? And how well would it perform if we limit the allowed permutations to ITG-like tree-transduction operations on the source parse tree? First we contribute oracle results showing great potential for performance improvement by source-reordering, ranging from 1.5 to 4 BLEU points depending on language pair. Although less outspoken, the potential of tree-based source-reordering is also significant. Our second contribution is a source reordering model that works with two kinds of tree transductions: the one permutes the order of sibling subtrees under a node, and the other first deletes layers in the parse tree in order to exploit sibling permutation at the remaining levels.The statistical parameters of the model we introduce concern individual tree transductions *conditioned on contextual features* of the tree resulting from all preceding transductions. Experiments in translating from English to Spanish/Dutch/Chinese show significant improvements of respectively 0.6/1.2/2.0 BLEU points.

1 Motivation

Word order differences between languages are a major challenge in Machine Translation (MT). Phrase-based Statistical Machine Translation (PBSMT) (Och and Ney, 2004; Zens et al., 2002; Koehn et al., 2003) deals with word order differences in two subcomponents of a translation model. Firstly, using the local word reordering implicitly encoded in phrase pairs. Secondly, using an explicit reordering model which may reorder target phrases relative to their source sides, e.g., as a monotone phrase sequence generation process with the possibility of swapping neighboring phrases (Tillman, 2004).

Arguably, local phrase reordering models cannot account for long-range reordering phenomena, e.g., (Chiang, 2005; Chiang, 2007). Hierarchical models of phrase reordering employ synchronous grammars or tree transducers, e.g., (Wu and Wong, 1998; Chiang, 2005). These models explore a more varied range of reordering phenomena, e.g., defined by at most inverting the order of sibling subtrees under each node in binary source/target trees (akin to ITG (Wu and Wong, 1998)).

Undoubtedly, the word order of source and target sentences is intertwined with the lexical choices on both side. Statistically speaking, however, one may first select a target word order given the source only, and then choose target words given the selected target word order and source words. One application of this idea is known as source reordering (or -permutation), e.g., (Collins et al., 2005; Xia and McCord, 2004; Wang et al., 2007; Li et al., 2007; Khalilov and Sima'an, 2010). Briefly, the words of the source string s are reordered to minimize word order differences with the target string t, leading to the source permuted string $\grave{s}$. Presumably, a standard PBSMT system trained to translate from $\grave{s}$ to t should have an easier task than translating directly from s to t. The source reordering part, s to $\grave{s}$,

[†] Currently, the first author is employed by TAUS B.V., Amsterdam (The Netherlands).

Proceedings of the 5th International Joint Conference on Natural Language Processing, pages 38–46,
Chiang Mai, Thailand, November 8 – 13, 2011. ©2011 AFNLP

can be realized in various ways and may manipulate morpho-syntactic parse trees of s, e.g., (Collins et al., 2005; Xia and McCord, 2004; Li et al., 2007).

It may seem that source reordering should provide only limited improvement over the standard PB-SMT approach. The literature reports mixed performance improvements for different language pairs, e.g., (Collins et al., 2005; Xia and McCord, 2004; Wang et al., 2007; Li et al., 2007; Khalilov and Sima'an, 2010). But what is the *potential improvement* of source reordering? We contribute experiments measuring oracle performance improvement for English to Dutch/Spanish/Chinese translations. Beside string-driven oracles, we report results using ITG-like transductions over a single syntactic parse tree of s. Our results confirm that reordering a single syntactic tree could be insufficient (e.g., (Huang et al., 2009)), yet they show substantial potential.

Our second contribution is a novel source reordering model that manipulates the source parse tree with two kinds of tree transduction operators: the one permutes the order of sibling subtrees under a node, and the other first abolishes layers in the parse tree in order to exploit sibling permutation at the remaining levels. The latter is the opposite of parse binarization using Expectation-Maximization (Huang et al., 2009). We use Maximum-Entropy training (Berger et al., 1996) to learn a sequence of tree transductions, each conditioned on contextual features *in tree resulting from outcome of the preceding transduction.* The conditioning on the outcome of preceding transductions is a departure from earlier approaches at learning independent source permutation steps, e.g., (Tromble and Eisner, 2009; Visweswariah et al., 2010).

The aim for the rest of this paper is firstly, to quantify the potential performance improvement of a standard PBSMT system if preceded by source reordering and secondly, to show that statistical Markov approach to tree transduction, where the probability of each transduction step is conditioned on the outcome of preceding steps, can improve the quality of PBSMT output significantly.

2 Source-Reordering: Framework

We start out from a word-aligned parallel corpus, consisting of triples $\langle s, a, t\rangle$, a source s, target t and word alignment a. Source reordering assumes that a permutation of s, called $\grave{s}$, is first generated with a model $P_r(\grave{s} \mid s)$ followed by a phrase translation model $P_t(t \mid \grave{s})$. The desired permutation $\grave{s}$ is one that has minimum word order divergence from t, i.e., when word-aligned again with t would have least number of crossing alignments.

Practically, the original parallel corpus $\{\langle s, a, t\rangle\}$ is split to two parallel corpora: (1) a source-to-permutation parallel corpus (consisting of $\langle s, a, \grave{s}\rangle$) and (2) a permutation-to-target parallel corpus (consisting of $\langle g\grave{s}, \grave{a}, t\rangle$), where $g\grave{s}$ is the output of a source reordering model (guessing at $\grave{s}$), and $\grave{a}$ results from automatically word aligning $\langle g\grave{s}, t\rangle$. The latter parallel corpus is used for training a phrase-based translation system $P_t(t \mid g\grave{s})$, while the former corpus is used for training a source reordering model $P_r(\grave{s} \mid s)$. The problem of permuting the source string to unfold the crossing alignments is computationally intractable (see (Tromble and Eisner, 2009)). However, various constraints can be made on unfolding the crossing alignments in a. A common approach is to assume a binary parse tree for the source string, and define a set of eligible permutations by binary ITG transductions. This defines permutations resulting from at most inverting pairs of children under nodes of the source tree. Figure 1 exhibits a long-range reordering of the verb in English-to-Dutch translation: inverting the order of the children of the VP node would unfold the crossing alignment. However, crossing alignments represented as non-constituents cannot be resolved. This difficulty can be circumvented by employing multiple alternative parse trees, by applying heuristic transforms (e.g., binarization) to the tree to fit the alignments (Wang et al., June 2010), or by defin-

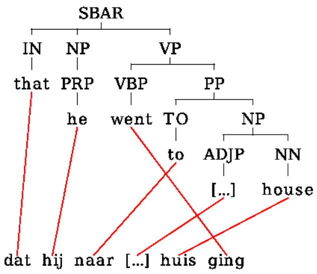


Figure 1: Example crossing alignments and long-distance reordering using a source parse tree.

ing new local transductions, on top child permutation (ITG) as we do next.

3 Existing work on source permutation

Source reordering has been shown useful for PBSMT for a wide variety of language pairs with high mutual word order disparity (Collins et al., 2005; Popovic' and Ney, 2006; Zwarts and Dras, 2007; Xia and McCord, 2004). In Costa-jussà and Fonollosa (2006) statistical word classes as well as POS tags are used as patterns for reordering the input sentences and producing a new bilingual pair.

A rather popular class of source reordering algorithms involves syntactic information and aims at minimizing the need for reordering during translation by permuting the source sentence (Collins et al., 2005; Wang et al., 2007; Khalilov and Sima'an, 2010; Li et al., 2007). Some systems perform source permutation using a set of handcrafted rules (Collins et al., 2005; Wang et al., 2007; Ramanathan et al., 2008), others make use of automatically learned reordering patterns extracted from the plain training data, the corresponding parse or dependency trees and the alignment matrix (Visweswariah et al., 2010).

Inspiring this work, source reordering as a pretranslation step is viewed as a word permutation learning problem in Tromble and Eisner (2009) and Li et al. (2007). The space of permutations is approached efficiently using a binary ITG-like synchronous context-free grammar put on the parallel data. Similarly, a local ITG-based tree transducer with contextual conditioning is used in Khalilov and Sima'an (2010) and Li et al. (2007), and preliminary experiments on a single language pair show improved performance.

Particularly, the model in (Li et al., 2007) is explicitly aimed at long-distance reorderings (English-Chinese), prunes the alignment matrix gradually to fit the source syntactic parse and employs Maximum-Entropy modeling to choose the optimal local ITG-like permutation step of sister subtrees but interleaves that step with a translation step. The model which we present in Section 2 differs substantially from (Li et al., 2007) and other earlier work because it (1) incorporates other kinds of tree transduction operations than those promoted by ITG, and (2) works with the unmodified alignment matrix but learns reorderings only from those alignments that are consistent with the tree, thereby avoiding the effects of heuristics for pruning alignments to fit the tree-structure, e.g., (Li et al., 2007).

In this paper we take the idea of learning source permutation one step further along a few dimensions. We show the utility of other kinds of tree transduction operations, besides those promoted by ITG, stress the importance of using a wide range of conditioning context features during learning, and report oracle and test results on *three* language pairs.

The majority of existing work reports encouraging performance improvements by source reordering. Next we aim at quantifying the potential improvement by oracle source reordering at the string level, if all permutations were to be allowed, and at the source syntactic tree level, by limiting the permutations with two kinds of local transductions.

4 Oracle source reordering results

Source reordering for PBSMT assumes that permuting the source words to minimize the order differences with the target sentence could improve translation performance. However, the question “how much?" is rarely asked. Here, we attempt answering this question with a set of oracle systems[1], in which we perform unfolding operations on the crossing links in alignment a (estimated between corpora s and t) that leads to a more monotone alignment $\grave{a}$ (between $\grave{s}$, which is a permutation of s, and t). We introduce a set of tree-based constraints that control the unfolding of alignment crossings. We measure the impact of (un)folded alignment crossings on the performance of the PBSMT system (see Table 1).

Oracle String. This method scans the alignment a from left-to-right and unfolds all the crossing links between bilingual phrases (*Oracle string*). Figure 2 shows an example of word reordering done on the string level. NULL aligned words do not move from their positions.

Oracle parse tree with permute siblings. The oracle system unfolds an alignment crossing if and

[1] All the source permutation methods presented in this Section are based on automatic alignments, which inevitably contain wrong links. In the future we plan to involve manual alignments to the computation of oracle permutation

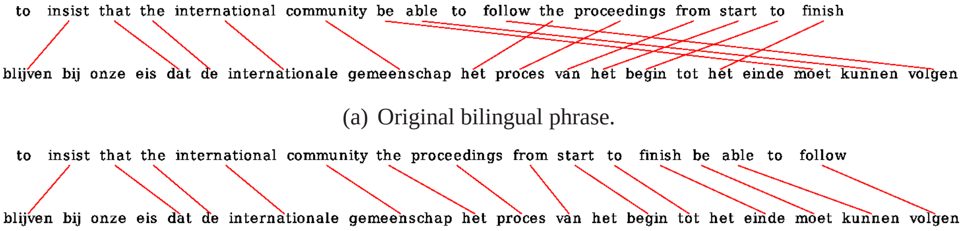
only if the source side of the alignment crossing is covered by the same node in the syntactic source tree, and the alignment pair subject to crossing can be unfolded by permuting the order of the sibling nodes. NULL aligned words do not prevent unfolding crossings because we include them with the adjacent words that are involved in the crossings. We call this configuration *Oracle tree*.

Figure 1 shows an example. According to the *Oracle tree* constraint, the word "went" can be placed in the end of the sentence since the replacement can be done as a swapping of "VBP" and "PP" categories. The same happens for the word "reflect" swapping with "S" constituent in Figure 1, but not for the chunks "the positions" and "not properly": this crossing cannot be resolved under the tree constraint since they are not dominated by sibling syntactic categories.

Oracle tree with delete descendants, permute siblings. This oracle implements an additional mechanism of tree modification to increase the number of reordering permutations in comparison with *Oracle tree* algorithm. Here we allow for an additional tree transduction operation that deletes intervening layers before applying sibling permutation. This is illustrated in Figure 3. The word "must" can not be moved to the beginning of the sentence in Figure 3a by *Oracle tree*. Instead, this is done in two steps. Firstly, the VP dominating the words "must" and "apply" is deleted under the current node S, the transformed tree is shown in Figure 3b. Subsequently, the siblings under S in the resulting tree are permuted, "must" is reordered across the whole clause and placed to the first position (see Figure 3c). We call this system *Oracle mod*.
Figure 4 shows an example in which crossing alignment links cannot be unfolded without deleting 4 intervening layers.

Oracle results Table 1 contrasts the oracle results with the performance shown by standard PBSMT systems. The experimental setup is detailed in Section 6. We consider the following baseline configurations: *PBSMT* - Moses-based PBSMT with distance-based reordering model; *PBSMT+MSD* - Moses-based PBSMT with distance-based reordering model and MSD and *Moses-chart* - hierarchical Moses-chart-based PBSMT.

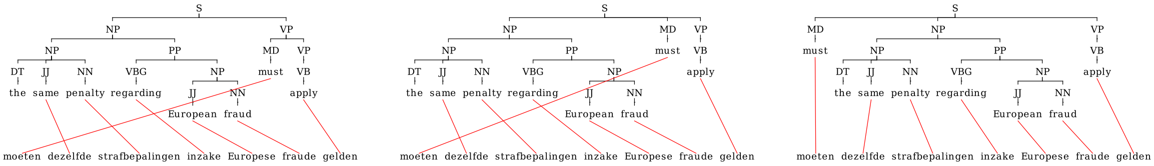
Depending on number of parse tree levels allowed to be deleted, we consider three *Oracle mod* systems: with two (*2lt*), three (*3lt*) and five (*5lt*) levels of descendants allowed to be deleted for a more flat parse tree structure before sibling permutation.


to insist that the international community be able to follow the proceedings from start to finish
blijven bij onze eis dat de internationale gemeenschap het proces van het begin tot het einde moet kunnen volgen
to insist that the international community the proceedings from start to finish be able to follow
blijven bij onze eis dat de internationale gemeenschap het proces van het begin tot het einde moet kunnen volgen

(a) Original bilingual phrase.

(b) Reordered bilingual phrase.

Figure 2: Example of *Oracle string* unfolding.


S
NP
VP
NP
PP
MD
VP
DT
JJ
NN
VBG
NP
must
VB
the same penalty regarding
JJ
NN
apply
European fraud
moeten dezelfde strafbepalingen inzake Europese fraude gelden
S
NP
MD
VP
NP
PP
must
VB
DT
JJ
NN
VBG
NP
apply
the same penalty regarding
JJ
NN
European fraud
moeten dezelfde strafbepalingen inzake Europese fraude gelden
S
MD
NP
VP
must
NP
PP
VB
DT
JJ
NN
VBG
NP
apply
the same penalty regarding
JJ
NN
European fraud
moeten dezelfde strafbepalingen inzake Europese fraude gelden

(a) Original parse tree.

(b) Parse tree with deleted VP category.

(c) Reordered parse tree.

Figure 3: Example of text monotonization with tree transformation.

The impact of corpus monotonization on translation system performance is measured using the final point of weight optimization on the development set (*Dev BLEU*), as well as on the test set (*Test BLEU/NIST*).

The major conclusion that can be drawn from the oracle results is that the source reordering defined in terms of parse tree transduction can potentially lead to increased translation quality (up to 1.2 BLEU points for English-Dutch, 0.5 for English-Spanish and 1.7 for English-Chinese). At the same time, a huge gap between performance shown by *Oracle string* and tree oracle systems ($\approx$2.2 BLEU points for English-Dutch, $\approx$1.3 for English-Spanish and $\approx$2.5 for English-Chinese) shows that there are many crossing alignments which cannot be unfolded with simple, local transductions over a single source-side syntactic tree.

5 Conditional Tree-Transductions

Our model aims at learning from the source permuted parallel corpus (containing tuples $\langle s, a, \grave{s}\rangle$) a probabilistic optimization ($\arg\max_{\pi(s)} P_r(\pi(s) \mid s, \tau_s)$), where τ_s is the source parse and $\pi(s)$ is some eligible permutation of s. We view the permutations leading from s to $\grave{s}$ as a sequence of local tree transductions $\tau_{\grave{s}_0} \to \ldots \to \tau_{\grave{s}_n}$, where $\grave{s}_0 = s$ and $\grave{s}_n = \grave{s}$, and each transduction $\tau_{\grave{s}_{i-1}} \to \tau_{\grave{s}_i}$ is defined using any of two kinds of local tree transduction operations used in Section 4 or alternatively NOP (No Operation).

The sequence $\tau_{\grave{s}_0} \to \ldots \to \tau_{\grave{s}_n}$ is obtained by taking the next node in a top-down tree traversal, then statistically selecting the most likely of three transduction operations and applying the selected operation to the current node. If the current tree is $\tau_{\grave{s}_{i-1}}$, and the current node has address x, is syntactically labeled N_x, directly dominates α_x (the ordered sequence of node labels under x), we approximate the conditional probability $P(\tau_{\grave{s}_i} \mid \tau_{\grave{s}_{i-1}})$ with the transduction operation it employs:

- Permute the children of x in $\tau_{\grave{s}_{i-1}}$ with probability

$$\approx P(\pi(\alpha_x) \mid N_x \to \alpha_x, C_x) \qquad (1)$$

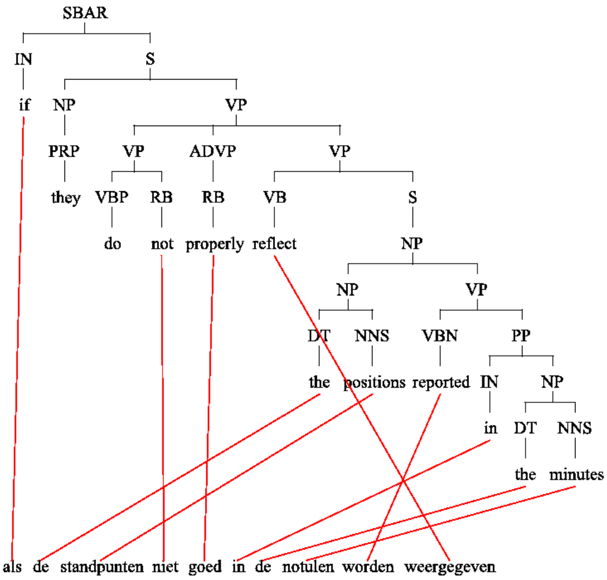


Figure 4: Example of unfoldable alignment crossings.

where $\pi(\alpha_x)$ is a permutation of α_x (the ordered sequence of node labels under x) and C_x is a local tree context of node x in tree $\tau_{s'_{i-1}}$.

- Select a child of x to delete, pull its children up directly under x, effectively changing α_x to some α_x^d, and then permute the children of the latter.

$$\approx P(\alpha_x \rightsquigarrow \alpha_x^d, \pi(\alpha_x) \mid N_x \to \alpha_x, C_x) \qquad (2)$$

where ($\alpha_x \rightsquigarrow \alpha_x^d$ symbolizes the result of deleting a subtree under a child of x. This operation applies also to subtrees of depth $n \in \{1, 2, 3, 5\}$ under x, i.e., a child is depth 1, a child with its children is depth 2 and so on.

Obviously, the number of possible permutations of α_x is factorial in the length of α_x. Fortunately, the source permuted training data exhibits only a fraction of possible permutations even for longer α_x sequences. Furthermore, by conditioning the probability on local context C_x, the number of permutations is limited to a handful set.

Theoretically, we could define the probability of the sequence of local tree transductions $\tau_{\grave{s}_0} \to \ldots \to \tau_{\grave{s}_n}$ as

$$P(\tau_{\grave{s}_0} \to \ldots \to \tau_{\grave{s}_n}) = \prod_{i=1}^{n} P(\tau_{\grave{s}_i} \mid \tau_{\grave{s}_{i-1}}) \qquad (3)$$

System	EnNl			EnEs			EnZh		
	Dev	Test		Dev	Test		Dev	Test	
	BLEU	BLEU	NIST	BLEU	BLEU	NIST	BLEU	BLEU	NIST
Baselines									
PBSMT	23.88	24.04	6.29	32.31	31.70	7.48	18.71	22.21	5.28
PBSMT+MSD	24.07	24.04	6.28	32.45	31.85	7.47	18.99	21.18	5.30
Moses-chart	23.94	24.93	6.39	30.58	31.80	7.41	19.93	23.90	5.41
Oracle results									
Oracle tree	24.70	24.80	6.32	32.76	32.21	7.51	20.23	23.44	5.35
Oracle string	26.28	27.02	6.50	34.09	33.52	7.60	23.01	26.08	5.52
Oracle mod+2lt	25.05	25.04	6.36	32.24	32.18	7.51	20.64	23.75	5.37
Oracle mod+3lt	25.11	25.27	6.37	32.22	32.34	7.52	20.71	23.59	5.37
Oracle mod+5lt	25.07	25.23	6.37	32.51	32.37	7.55	20.93	23.93	5.39

Table 1: Summary of oracle results.

However, unlike earlier work (e.g., (Tromble and Eisner, 2009)), we cannot afford to do so because every local transduction conditions on context C_x of an intermediate tree, which quickly risks becoming intractable (even when we use packed forests). Furthermore, the problem of calculating the most likely permutation under such a model is made difficult by the fact that different transduction sequences may lead to the same permutation, which demands summing over these sequences (another intractable summation). Earlier work has avoided conditioning context, effectively assuming that the each intermediate permutation is independent from the preceding ones.

Instead, we take a pragmatic approach and greedily select at every intermediate point $\tau_{\grave{s}_{i-1}} \rightarrow \tau_{\grave{s}_i}$ the single most likely local transduction that can be applied to a node in the current intermediate tree $\tau_{\grave{s}_{i-1}}$ using an interpolation of the terms in Equations 1 and 2 with probability ratios of the language model $\grave{s}$ as follows:

$$P(TRANS_i|\, N_x \rightarrow \alpha_x, C_x) \times \frac{P_{lm}(\grave{s}_{i-1})}{P_{lm}(\grave{s}_i)}$$

where $TRANS_i$ is any of the two transduction operations or NOP, and P_{lm} is a language model trained on the $\grave{s}$ side of the corpus $\{\langle s, a, \grave{s}\rangle\}$. The rationale behind this log-linear interpolation is that our source permutation approach aims at finding the optimal permutation $\grave{s}$ 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.

Estimates. We estimate the string probabilities $P_{lm}(.)$ using 3-gram language models trained on the $\grave{s}$ side of the source permuted parallel corpus $\{\langle s, a, \grave{s}\rangle\}$. We estimate the conditional probability $P_r(TRANS \mid N_x \rightarrow \alpha_x, C_x)$ 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 α_x together with the corresponding sequence of head POS tags and other features corresponding to different contextual information.

Features in use. We used a set of 15 features to capture reordering permutations from the syntactic and linguistic perspectives: *Local tree topology:* sub-tree instances that include parent node and the ordered sequence of child node labels (1); *Dependency features:* features that determine the POS tag of the head word of the current node (2), together with the sequence of POS tags of the head words of its child nodes (3) and the POS tag of the head word of the parent (4) and grandparent nodes (5); *Syntactic features:* apart from the whole path from the current node to the tree root node (6), we used three binary features from this class describe: (7) whether the parent node is a child of the node annotated with the same syntactic category, (8) whether the parent node is a descendant of the node annotated with the same syntactic category, and (9) if the current subtree is embedded into a "*S-SBAR*" sub-tree[2]; *POS lexical features:* bi- and tri-grams of POS tags of the left- and right-hand side neighboring words (10-13); *Counters:* a number of words covered by a given constituent (14) and a number of children of the given node (15).

6 Translation and reordering experiments

Data. In our experiments we used English-Dutch and English-Spanish European Parliament data and an extraction from the English-Chinese Hong Kong Parallel Corpus. All the sets were provided with one reference translation. Basic statistics of the training data can be found in Table 2, development datasets contained 0.5K, 1.9K and 0.5K lines and test datasets contained 1K, 1.9K and 0.5K for Dutch, Spanish and English, respectively.

Experimental setup. Word alignment was found using GIZA++[3] (Och, 2003), supported by mkcls[4] (Och, 1999) tool. The PBSMT systems we consider in this study is based on Moses toolkit (Koehn et al., 2007). We followed the guidelines provided on the Moses web page[5].

Two phrase reordering methods are widely used in phrase-based systems. A distance-based reordering model providing the decoder with a cost linear to the distance between words that are being reordered. This model constitutes the default for the Moses system. And, a lexicalized block-oriented data-driven reordering model (Tillman, 2004) considers three orientations: monotone (M), swap (S), and discontinuous (D), while the reordering probabilities are conditioned on the lexical context of each phrase pair.

All language models were trained with SRI LM toolkit (Stolcke, 2002). Language models for Dutch, Spanish and Chinese use 5-grams, while the idealized English ($\grave{s}$) is modeled using 3-grams.

We use Stanford parser[6] (Klein and Manning, 2003) as a source-side parsing engine. The parser was trained on the WSJ Penn treebank 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[7].

Translation scores. Table 3 shows the results of automatic evaluation using BLEU (Papineni et al., 2002) and NIST (Doddington, 2002) metrics.

MERrd configuration corresponds to the PBSMT system with the source side of the parallel corpus reordered using our Maximum Entropy model, but the transduction operations are limited to permutation of the children only. *MERrd+xlt* configuration refers to the set of systems which, beside child permutation, includes a deletion operation with the maximum number of tree layers that can be deleted set to x. All reordered systems include a MSD model as a supporting reordering mechanism.

BLEU scores measured on the test data, which are statistically significant from the best PBSMT results are marked with bold. The statistical significance calculations have been done for a 95% confidence interval and 1000 resamples, following the guidelines in Koehn (2004).

Analysis. Our results show that source-reordering is beneficial for the language pairs with high mutual word order disparity. In contrast to English-Dutch and English-Chinese translation tasks, the statistical significance test reveals that all but the *MERrd+5lt* English-Spanish PBSMT systems with rearranged input are not different from the translation quality delivered by Moses. This disappointing result for the English-to-Spanish translation task may be explained by the fact that many reordering differences are resolved by standard reordering models (distance-based and MSD).

Table 3 shows the results of automatic translation quality evaluation. A gap between the maximum reachable performance shown by tree transduction systems and the translation quality delivered

[2] The latter feature intends to model the divergence in word order in relative clauses between Dutch and English which is illustrated in Figure 1.

[3] code.google.com/p/giza-pp/

[4] http://www.fjoch.com/mkcls.html

[5] http://www.statmt.org/moses/

[6] http://nlp.stanford.edu/software/lex-parser.shtml

[7] http://homepages.inf.ed.ac.uk/lzhang10/maxent_toolkit.html

Parameter	Dutch	English	Spanish	English	Chinese	English
Training corpus						
Sentences	1.2 M	1.2 M	1.4 M	1.4 M	1.5 M	1.5 M
Words	32.9 M	33.0 M	40.1 M	38.54 M	35.35 M	35.00 M
Vocabulary	228 K	104 K	168 K	119 K	136 K	245 K
Average sentence length	27.20	27.28	28.80	27.67	24.06	23.83

Table 2: Statistics of the training, development and test corpora.

by our model is 0.05-0.29 BLEU points for English-Dutch, 0.01-0.09 for English-Spanish and 0.27-0.76 for English-Chinese. These numbers demonstrate that there are some potentially usable regularities not captured by our current conditional tree-transduction model.

7 Conclusions and future work

We present a source reordering system for PBSMT, in which the reordering decisions are conditioned on features from the source parse tree. Our system allows for two operations over the parse tree: permuting the order of sibling nodes and deleting child nodes in order to make the tree flatter and exploit sibling permutations at the remaining layers.

Our contribution can be summarized as follows: (1) we report detailed results of maximum potential performance that can be achieved with source reordering under different constraints, (2) we define a source-reordering process through an efficient sequence of greedy, context-conditioned transduction operations over the source parse trees.

The method was tested on three different translation tasks. The results show that our approach is more effective for language pairs with significant difference in word order. Another important observation is that our model demonstrate translation quality comparable with the one delivered by SMT systems based on hierarchical phrases.

The introduced reordering algorithm and the results obtained present many opportunities for future work. We plan to perform a detailed analysis of the structure of the extracted phrases to find out the particular cases where the improvement comes from. We also propose to discover other possible transduction operations to better explore the reordering space.

8 Acknowledgements

This work is supported by The Netherlands Organization for Scientific Research (NWO) under VIDI grant (nr. 639.022.604).

System	EnNl			EnEs			EnZh		
	Dev	Test		Dev	Test		Dev	Test	
	BLEU	BLEU	NIST	BLEU	BLEU	NIST	BLEU	BLEU	NIST
Baselines									
PBSMT	23.88	24.04	6.29	32.31	31.70	7.48	18.71	22.21	5.28
PBSMT+MSD	24.07	24.04	6.28	32.45	31.85	7.47	18.99	21.18	5.30
Moses-chart	23.94	24.93	6.39	30.58	31.80	7.41	19.93	23.90	5.41
Reordering systems									
MERrd	24.64	**24.72**	6.33	31.97	32.19	7.52	19.82	**23.17**	5.33
MERrd+2lt	24.61	**24.99**	6.35	31.70	32.11	7.50	20.02	23.01	5.33
MERrd+3lt	24.82	**24.98**	6.34	31.65	32.25	7.52	20.21	**23.14**	5.34
MERrd+5lt	24.78	**25.12**	6.37	31.99	**32.38**	7.52	20.29	**23.17**	5.35

Table 3: Experimental results.

References

A. Berger, S. Della Pietra, and V. Della Pietra. 1996. A maximum entropy approach to natural language processing. *Computational Linguistics*, 1(22):39–72.

D. Chiang. 2005. A hierarchical phrase-based model for statistical machine translation. In *Proceedings of ACL'05*, pages 263–270, Ann Arbor, MI, USA.

D. Chiang. 2007. Hierarchical phrase-based translation. *Computational Linguistics*, 2(33):201–228.

M. Collins, P. Koehn, and I. Kučerová. 2005. Clause restructuring for statistical machine translation. In *Proceedings of ACL'05*, pages 531–540, Ann Arbor, MI, USA.

M. R. Costa-jussà and J. A. R. Fonollosa. 2006. Statistical machine reordering. In *Proceedings of HLT/EMNLP'06*, pages 70–76, New York, NY, USA.

G. Doddington. 2002. Automatic evaluation of machine translation quality using n-grams co-occurrence statistics. In *Proceedings of HLT'02*, pages 128–132, San Diego, CA, USA.

L. Huang, H. Zhang, D. Gildea, and K. Knight. 2009. Binarization of synchronous context-free grammars. *Computational Linguistics*, 35(4):559–595.

M. Khalilov and K. Sima'an. 2010. A discriminative syntactic model for source permutation via tree transduction. In *Proceedings of SSST-4 workshop at COLING'10*, pages 92–100, Beijing, China.

D. Klein and C. Manning. 2003. Accurate unlexicalized parsing. In *Proceedings of ACL'03*, pages 423–430, Sapporo, Japan.

Ph. Koehn, F. Och, and D. Marcu. 2003. Statistical phrase-based machine translation. In *Proceedings of the HLT-NAACL'03*, pages 48–54, Edmonton, Canada.

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'07*, pages 177–180, Prague, Czech Republic.

Ph. Koehn. 2004. Statistical significance tests for machine translation evaluation. In *Proceedings of EMNLP'04*, pages 388–395, Barcelona, Spain.

C. Li, L. Minghui, D. Zhang, M. Li, M. Zhou, and Y. Guan. 2007. A probabilistic approach to syntax-based reordering for statistical machine translation. In *Proceedings of ACL'07*, pages 720–727, Prague, Czech Republic.

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'99*, pages 71–76, Maryland, MD, USA.

F. Och. 2003. Minimum error rate training in statistical machine translation. In *Proceedings of ACL'03*, pages 160–167, Sapporo, Japan.

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, Philadelphia, PA, USA.

M. Popovic' and H. Ney. 2006. POS-based word reorderings for statistical machine translation. In *Proceedings of LREC'06*, pages 1278–1283, Genoa, Italy, May.

A. Ramanathan, P. Bhattacharyya, J. Hegde, R.M. Shah, and M. Sasikumar. 2008. Simple syntactic and morphological processing can help english-hindi statistical machine translation. In *In Proceedings of IJCNLP'08*, Hyderabad, India.

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, Boston, MA, USA.

R. Tromble and J. Eisner. 2009. Learning linear ordering problems for better translation. In *Proceedings of EMNLP'09*, pages 1007–1016, Singapore.

K. Visweswariah, J. Navratil, J. Sorensen, V. Chenthamarakshan, and N. Kambhatla. 2010. Syntax based reordering with automatically derived rules for improved statistical machine translation. In *Proceeding of COLING'10*, pages 1119–1127, Beijing, China.

C. Wang, M. Collins, and Ph. Koehn. 2007. Chinese syntactic reordering for statistical machine translation. In *Proceedings of EMNLP-CoNLL'07*, pages 737–745, Prague, Czech Republic.

W. Wang, J. May, K. Knight, and D. Marcu. June 2010. Re-structuring, re-labeling, and re-aligning for syntax-based machine translation. *Computational Linguistics*, 36:247–277.

D. Wu and H. Wong. 1998. Machine translation wih a stochastic grammatical channel. In *Proceedings of ACL-COLING'98*, pages 1408–1415, Columbus, OH, USA.

F. Xia and M. McCord. 2004. Improving a statistical MT system with automatically learned rewrite patterns. In *Proceedings of COLING'04*, pages 508–514, Geneva, Switzerland.

R. Zens, F. Och, and H. Ney. 2002. Phrase-based statistical machine translation. In *Proceedings of KI: Advances in Artificial Intelligence*, pages 18–32.

S. Zwarts and M. Dras. 2007. Syntax-based word reordering in phrase-based statistical machine translation: Why does it work? In *Proceedings of the MT Summit XI*, Copenhagen, Denmark.