Examining the Relationship between Preordering and Word Order Freedom in Machine Translation

Joachim Daiber **Miloš Stanojević** **Wilker Aziz** **Khalil Sima'an**
Institute for Logic, Language and Computation (ILLC)
University of Amsterdam
{initial.last}@uva.nl

Abstract

We study the relationship between word order freedom and preordering in statistical machine translation. To assess word order freedom, we first introduce a novel entropy measure which quantifies how difficult it is to predict word order given a source sentence and its syntactic analysis. We then address preordering for two target languages at the far ends of the word order freedom spectrum, German and Japanese, and argue that for languages with more word order freedom, attempting to predict a unique word order given source clues only is less justified. Subsequently, we examine lattices of n-best word order predictions as a unified representation for languages from across this broad spectrum and present an effective solution to a resulting technical issue, namely how to select a suitable source word order from the lattice during training. Our experiments show that lattices are crucial for good empirical performance for languages with freer word order (English–German) and can provide additional improvements for fixed word order languages (English–Japanese).

1 Introduction

Word order differences between a source and a target language are a major challenge for machine translation systems. For phrase-based models, the number of possible phrase permutations is so large that reordering must be constrained locally to make the search space for the best hypothesis feasible. However, constraining the space locally runs the risk that the optimal hypothesis is rendered out of reach. Preordering of the source sentence has been embraced as a way to ensure the reachability of certain target word order constellations for improved prediction of the target word order. Preordering aims at predicting a permutation of the source sentence which has minimal word order differences with the target sentence; the permuted source sentence is passed on to a backend translation system trained to translate target-order source sentences into target sentences. In essence, the preordering approach makes the assumption that it is feasible to predict target word order given only clues from the source sentence. In the vast majority of work on preordering, a single preordered source sentence is passed on to the backend system, thereby making the stronger assumption that it is feasible to predict a *unique* preferred target word order. But how reasonable are these assumptions and for which target languages?

Intuitively, the assumption of a *unique preordering* seems reasonable for translating into fixed word order languages such as Japanese, but for translation into languages with less strict word order such as German, this is unlikely to work. In such languages there are multiple comparably plausible target word orders per source sentence because the underlying predicate-argument structure can be expressed with mechanisms other than word order alone (e.g. morphological inflections or intonation). For these languages, it seems rather unlikely to be able to choose a unique word order given only source sentence clues. In this paper, we want to shed light on the relationship between the target language's word order freedom and the feasibility of preordering. We start out by contributing an information-theoretic measure to quantify the difficulty in predicting a preferred word order given the source sentence and its syntax. Our measure provides empirical support for the intuition that it is often not possible to predict a unique word order for free word order languages, whereas it is

Proceedings of the First Conference on Machine Translation, Volume 1: Research Papers, pages 118–130,
Berlin, Germany, August 11-12, 2016. ©2016 Association for Computational Linguistics

more feasible for fixed word order languages such as Japanese. Subsequently, we study the option of passing the n-best word order predictions, instead of 1-best, to the backend system as a lattice of possible word orders of the source sentence.

For the training of the backend system, the use of such permutation lattices raises a question: What should constitute the training corpus for a lattice-preordered translation system? In previous work using single word order predictions, the training data consists of pairs of source and target sentences where the source sentence is either in target order (i.e. order based on word alignments) or preordered (i.e. predicted order). In this work we contribute a novel approach for selecting training instances from the lattice of word order permutations: We select the permutation providing the best match with the target-order source sentence (we call this process "lattice silver training").

Our experiments show that for English–Japanese and English–German lattice preordering has a positive impact on the translation quality. Whereas lattices enable further improvement for preordering English into the strict word order language Japanese, lattices in conjunction with our proposed lattice silver training scheme turn out to be crucial to reach satisfactory empirical performance for English–German. This result highlights that when predicting word order of free word order languages given source clues only, it is important to ensure that the word order predictions and the backend system are suitably fitted together.

2 Related Work

Preordering has been explored from the perspective of the upper-bound achievable translation quality in several studies, including Khalilov and Sima'an (2012) and Herrmann et al. (2013), which compare various systems and provide oracle scores for syntax-based preordering models. Target-order source sentences, in which the word order is determined via automatic alignments, enable translation systems great jumps in translation quality and provide improvements in compactness and efficiency of downstream phrase-based translation models. Approaches have largely followed two directions: (1) predicting word order based on some form of source-syntactic representation and (2) approaches which do not depend on source syntax.

2.1 Source Syntax-Based Preordering

Many approaches to preordering have made use of syntactic representations of the source sentence, including Collins et al. (2005) who restructure the source phrase structure parse tree by applying a sequence of transformation rules. More recently, Jehl et al. (2014) learn to order sibling nodes in the source-side dependency parse tree. The space of possible permutations is explored via depth-first branch-and-bound search (Balas and Toth, 1983). In later work, the authors further improve this model by replacing the logistic regression classifier with a feed-forward neural network (de Gispert et al., 2015), which results in improved empirical results and eliminates the need for feature engineering. Lerner and Petrov (2013) train classifiers to predict the permutations of up to 6 tree nodes in the source dependency tree. The authors found that by only predicting the best 20 permutations of n nodes, they could cover a large majority of the reorderings in their data.

2.2 Preordering without Source Syntax

Tromble and Eisner (2009) learn to predict the orientation of any two words (straight or inverted order) using a perceptron. The search for the best reordering is performed with a $O(n^3)$ chart parsing algorithm. More basic approaches to syntax-less preordering include the application of multiple MT systems (Costa-jussà and Fonollosa, 2006), where a first system learns preordering and a second learns to translate the preordered sentence into the target sentence. Finally, there have been successful attempts at the automatic induction of parse trees from aligned data (DeNero and Uszkoreit, 2011) and the estimation of latent reordering grammars (Stanojević and Sima'an, 2015) based on permutation trees (Zhang and Gildea, 2007).

2.3 Lattice Translation

A lattice is an acyclic finite-state automaton defining a finite language. A more restricted class of lattices, namely, confusion networks (Bertoldi et al., 2007), has been extensively used to pack alternative input sequences for decoding.[1] However, applications mostly focused on speech translation (Ney, 1999; Bertoldi et al., 2007), or to account for lexical and/or segmentation ambiguity due to preprocessing (Xu et al., 2005; Dyer, 2007). In very

[1] A confusion network is a special case of a lattice where every path from start to final state goes through every node.

few occasions, lattice input has been used to determine the space of permutations of the input considered by the decoder (Knight and Al-Onaizan, 1998; Kumar and Byrne, 2003). The effectiveness of lattices of permutations was demonstrated by Zhang et al. (2007). However, except in the cases of n-gram based decoders (Khalilov et al., 2009) this approach is not a common practice.

Dyer et al. (2008) formalized lattice translation both for phrase-based and hierarchical phrase-based MT. The former requires a modification of the standard phrase-based decoding algorithm as to maintain a coverage vector over states, rather than input word positions. The latter requires intersecting a lattice and a context-free grammar, which can be seen as a generalized form of parsing (Klein and Manning, 2001). In this work, we focus on phrase-based models.

The space of translation options in standard phrase-based decoding with a distortion limit d grows with $O(\text{stack size} \times n \times 2^d)$ where n represents the input length, and the number of translation options is capped due to beam search (Koehn et al., 2003). With lattice input, the dependency on n is replaced by $|Q|$ where Q is the set of states of the lattice. The *stack size* makes the number of translation options explored by the decoder independent of the number of transitions in the lattice.

As in standard decoding, the states of a lattice can also be visited non-monotonically. However, two states in a lattice are not always connected by a path, and, in general, paths connecting two nodes might differ in length. Dyer et al. (2008) proposed to pick the shortest path between two nodes to be representative of the distance between them.[2] Just like in standard decoding, a *distortion limit* is imposed to keep the space of translations tractable.

In this work, we use lattice input to constrain the space of permutations of the source allowed within the decoder. Moreover, in most cases we completely disable the decoder's further reordering capabilities. Because our models can perform global permutation operations without ad hoc distortion limits, we can reach far more complex word orders. Crucially, our models are better predictors of word order than standard distortion-based reordering, thus we manage to decode with relatively small permutation lattices.

3 Quantifying Word Order Freedom

While varying degrees of word order freedom are a well-studied topic in linguistics, word order freedom has only recently been studied from a quantitative perspective. This has been enabled partly by the increasing availability of syntactic treebanks. Kuboň and Lopatková (2015) propose a measure of word order freedom based on a set of six common word order types (SVO, SOV, etc.). Futrell et al. (2015) define various entropy measures based on the prediction of word order given unordered dependency trees. Both approaches require a dependency treebank for each language.

In practical applications such as machine translation, it is difficult to quantify the influence of word order freedom. For an arbitrary language pair, our goal is to quantify a notion of the target language's word order freedom based only on parallel sentences and source syntax. In their head direction entropy measure, Futrell et al. (2015) approach the problem of quantifying word order freedom by measuring the difficulty of recovering the correct linear order from a sentence's unordered dependency tree. We approach the problem of quantifying a target language's word order freedom by measuring the difficulty of predicting target word order based on the source sentence's dependency tree. Hence, we ask questions such as: How difficult is it to predict French word order based on the syntax of the English source sentence?

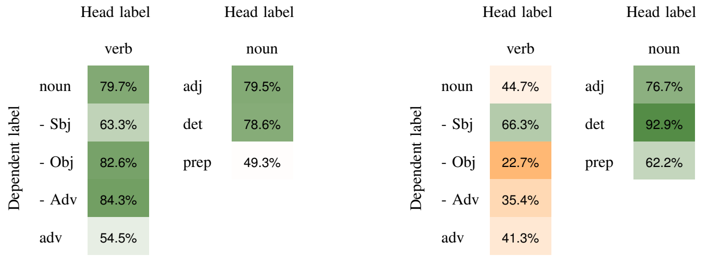
3.1 Source Syntax and Target Word Order

We represent the target sentence's word order as a sequence of order decisions. Each order decision encodes for two source words, a and b, whether their translation equivalents are in the order (a, b) or (b, a). The source sentences are parsed with a dependency parser.[3] The target-language order of the words in the source dependency tree is then determined by comparing the target sentence positions of the words aligned to each source word. Figure 1 shows the percentage of dependent-head pairs in the source dependency tree whose target order can be correctly guessed by always choosing the more common decision.[4]

[2] This is achieved by running an all-pairs shortest path algorithm prior to decoding – see for example Chapter 25 of (Cormen et al., 2001). MOSES uses the Floyd-Warshall algorithm, which runs in time $O(|Q|^3)$.

[3] `http://cs.cmu.edu/~ark/TurboParser/`

[4] For English–Japanese, we use manual word alignments of 1,235 sentences from the *Kyoto Free Translation Task* (Neubig, 2011) and for English–German, we use a manually word-aligned subset of Europarl (Padó and Lapata, 2006) consisting of 987 sentences.



(a) English–Japanese

(b) English–German

Figure 1: Source word pairs whose target order can be predicted using only the words' labels.

German and Japanese Both language pairs differ significantly in how strictly the target language's word order is determined by the source language's syntax. English–German shows strict order constraints within phrases, such as that adjectives and determiners precede the noun they modify in the vast majority of cases (Figure 1b). However, English–German also shows more freedom on the clause level, where basic syntax-based predictions for the positions of nouns relative to the main verb are insufficient. For English–Japanese on the other hand, the position of the nouns relative to the main verb is more rigid, which is demonstrated by the high scores in Figure 1a. These results are in line with the linguistic descriptions of both target languages. From a technical point of view, they highlight that any treatment of English–German word order must take into account information beyond the basic syntactic level and must allow for a given amount of word order freedom.

3.2 Bilingual Head Direction Entropy

While such a qualitative comparison provides insight into the order differences of selected language pairs, it is not straight-forward to compare across many language pairs. From a linguistic perspective, Futrell et al. (2015) use entropy to compare word order freedom in dependency corpora across various languages. While the authors observed that artifacts of the data such as treebank annotation style can hamper comparability, they found that a simple entropy measure for the prediction of word order based on the dependency structure provided a good quantitative measure of word order freedom.

We follow Futrell et al. (2015) in basing our measure on conditional entropy, which provides a straight-forward way to quantify to which extent target word order is determined by source syntax.

$$H(Y|X) = -\sum_{x \in \mathcal{X}} p(x) \sum_{y \in \mathcal{Y}} p(y|x) \log p(y|x)$$

Conditional entropy measures the amount of information required to describe the outcome of a random variable Y given the value of a second random variable X. Given a dependent-head pair in the source dependency tree, X consists of the dependent's and the head's part of speech, as well as the dependency relation between them. Note that as in all of our experiments the source language is English, the space of outcomes of X is the same across all language pairs.Y in this case is the word pair's target-side word order in the form of a (a, b) or (b, a) decision. We estimate $H(Y|X)$ using the bootstrap estimator of DeDeo et al. (2013), which is less prone to sample bias than maximum likelihood estimation.[5]

Influence of word alignments Futrell et al. (2015) use human-annotated dependency trees for each language they consider. Our estimation only involves word-aligned bilingual sentence pairs with a source dependency tree. Manual alignments are available for a limited number of language pairs and often only for a diminishingly small number of sentences. Consequently the question arises, whether automatic word alignments are sufficient for this task. To answer this question, we apply our measure to a set of manually aligned as well as a larger set of automatically aligned sentence pairs. In addition to the German and Japanese alignments mentioned above, we use manual alignments for English–Italian (Farajian et al., 2014), English–French (Och and Ney, 2003), English–Spanish (Graça et al., 2008) and English–Portuguese (Graça et al., 2008).

[5] We observe an average of 1,033 values for X per language pair and perform 10,000 Monte-Carlo samples.

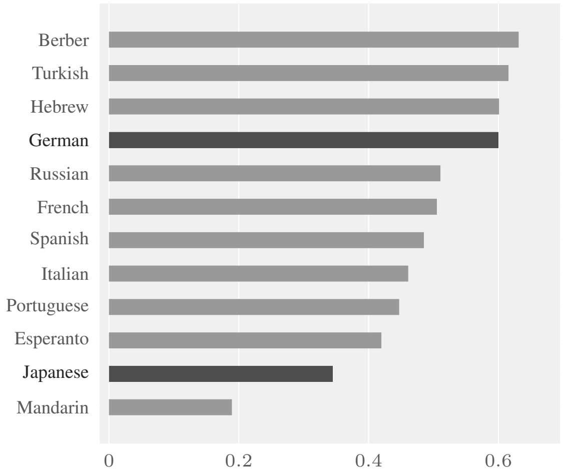


Figure 2: Bilingual head direction entropy with English source side.

Since a limited number of manually aligned sentences are available, it is important to avoid bias due to sample size. Hence, we randomly sample the same number of dependency relations from each language pair. Considering only those languages for which we have both manual and automatic alignments, we can determine how well their word order freedom rankings correlate. Even though the number of samples for the manually aligned sentences is limited to 500 due to the size of the smallest set of manual alignments, we find a high correlation of Spearman's $\rho = 0.77$ between the rankings of the 6 languages that occur in both sets (Zwillinger and Kokoska, 1999).

Influence of source syntax Another factor that may influence our estimated degree of word order freedom is the form and granularity of the source side's syntactic representation: More detailed representations may disambiguate cases that are difficult to predict with a more bare representation. As we are interested in the bilingual case and, specifically, in preordering, we content ourselves with using the same syntactic representation, i.e. dependency trees, that many preordering models use (e.g., Jehl et al. (2014), Lerner and Petrov (2013)).

Comparison to monolingual measures Our measure is similar to Futrell et al. (2015)'s head direction entropy; however, it also offers several advantages. While monolingual head direction entropy requires a dependency treebank for each language, our bilingual head direction entropy only requires dependency annotation for the source language (English in our case). One of their caveats, the influence of the widely varying dependency annotation styles across treebanks, is also not present in our method, since a single dependency style is used for the source language. We have demonstrated that automatic alignments perform on a comparable level to manual alignments. Accordingly, the amount of data that can be used to estimate the measure is only limited by the availability of parallel sentences. Finally, while dependency treebanks rarely cover the same corpora or even domains, our method can utilize sentences from the same or similar corpora for each language, thus minimizing potential corpus biases.

Translation from English Figure 2 plots bilingual head direction entropy for an English source side and a set of typologically diverse languages on the target side. For each language pair, we use 18,000 sentence pairs and automatic alignments from the Tatoeba corpus (Tiedemann, 2012).[6]

Languages at the top of the plot in Figure 2 show a greater degree of word order freedom with respect to the English source syntax. Thus, predicting their word order from English source clues alone is likely to be difficult. We argue that in such cases it is crucial to pass on the ambiguity over the space of predictions to the translation model. By doing so, word order decisions can be influenced by translation decisions, while still shaping the space of reachable translations.

4 Preordering Free and Fixed Word Order Languages

The measure of word order freedom introduced in the previous section enables us to estimate how difficult it is to predict the target language's word order based on the source language. In this section, we introduce the two preordering models we use to predict the word order of German and Japanese. Experiments with these models will allow us to examine the relationship between preordering and word order freedom.

4.1 Neural Lattice Preordering

Based on their earlier work, which used logistic regression and graph search for preordering (Jehl et al., 2014), de Gispert et al. (2015) introduce a neural preordering model. In this model, a feed-forward neural network is trained to estimate the

[6] The alignments were produced using GIZA++ (Och and Ney, 2003) with *grow-diag-final-and* symmetrization.

swap probabilities of nodes in the source-side dependency tree. Search is performed via the depth-first branch-and-bound algorithm. The authors have found this model to be fast and to produce high quality word order predictions for a variety of languages.

Model estimation Training examples are extracted from all possible pairs of children of a dependency tree node, including the head itself. For each pair, the two nodes are swapped if swapping them reduces the number of crossing alignment links. The crossing score of two nodes a and b (a precedes b in linear order) and their aligned target indexes A_a and A_b is defined as follows:

$$\mathrm{cs}(a,b) = |\,\{(i,j) \in A_a \times A_b : i > j\}\,|$$

Training instances generated in this manner are then used to estimate the swap probability $p(i,j)$ for two indexes i and j. For each node in the source dependency tree, the best possible permutation of its children (including the head) is determined via graph search. The score of a permutation of length k is defined as follows:

$$\mathrm{score}(\pi) = \prod_{1 \leq i < j \leq k | \pi[i] > \pi[j]} p(i,j) \cdot \prod_{1 \leq i < j \leq k | \pi[i] < \pi[j]} 1 - p(i,j)$$

We closely follow de Gispert et al. (2015) for the implementation of the estimator of $p(i,j)$. A feed-forward neural network (Bengio et al., 2003) is trained to predict the orientation of a and b based on a sequence of 20 features, such as the words, the words' POS tags, the dependency labels, etc.[7] The network consists of 50 nodes on the input layer, 2 on the output layer, and 50 and 100 on the two hidden layers. We use a learning rate of 0.01, batch size of 1000 and perform 20 training epochs.

Search Search in this model consists of finding the sequence of swaps leading to the best overall score according to the model. Let a partial permutation of k nodes be a sequence of length $k' < k$ containing each integer in $\{1, ..., k\}$ at most once. The score of a new permutation obtained by extending a partial permutation π' of length k' by one element can be computed efficiently as:

$$\mathrm{score}(\pi' \cdot \langle i \rangle) = \mathrm{score}(\pi') \cdot \prod_{j \in V | i > j} p(i,j) \cdot \prod_{j \in V | i < j} 1 - p(i,j)$$

k-best search Target languages such as German allow for a significant amount of word order freedom; hence, the depth-first branch-and-bound algorithm, which extracts the single best permutation, may not be the best choice in this case. In the context of the Traveling Salesman Problem, van der Poort et al. (1999) show that general branch-and-bound search can be extended to retrieve k-best results while keeping the same guarantees and computational complexity. Only minor changes are necessary to adapt the search for the best permutation to finding the k-best permutations: We keep a set $bestk$ of the best permutations and a single $bound$. If for a permutation π', $\mathrm{score}(\pi') > bound$, instead of updating the bound to the single best permutation and remembering it, the following steps are performed:

1. If $|bestk| = k$:
   - Remove worst permutation from the set.
2. Add π' to $bestk$.
3. The new $bound$ will be the score of the worst permutation in $bestk$.

4.2 Reordering Grammar Induction

Reordering Grammar (RG) (Stanojević and Sima'an, 2015) is a recent approach for preordering that is hierarchical and fully unsupervised. It is based on inducing a probabilistic context-free grammar from aligned parallel data. This grammar can predict permutation trees (PETs) (Zhang and Gildea, 2007) — projective constituency trees that can fully describe any permutation. PETs are reminiscent of ITG (Wu, 1997) with the important distinction that PETs can handle any permutation, unlike ITG which can only handle binarizable ones. As in ITG, constituents in PETs are labeled with the permutation of their children.

Induction of RGs is performed by specifying a generative probabilistic model and then estimating its parameters using the EM algorithm. The reasoning behind using EM is that many latent variables are present in the model. Only the source

[7] Our implementation is based on `http://nlg.isi.edu/software/nplm/`.

sentence and its permutation are observed during training. The exact PET that generated this permutation is not observed and there could be (exponentially) many PETs that could have generated the observed permutation. Hence, the bracketings of potential PETs are treated as latent variables.

The second source of latent variables is state splitting of non-terminals (labels that indicate how to reorder the children) in a similar way as done in monolingual parsing (Matsuzaki et al., 2005; Petrov et al., 2006; Prescher, 2005). Each latent permutation tree has many latent derivations and the generative probabilistic model needs to account for them. The probability of the observed permutation π is defined in the following way:

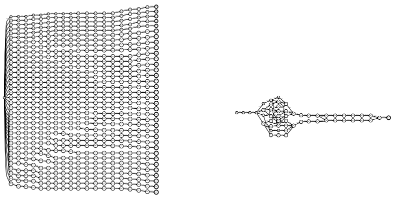
$$P(\pi) = \sum_{\Delta \in \mathrm{PEF}(\pi)} \sum_{d \in \Delta} \prod_{r \in d} P(r)$$

where $\mathrm{PEF}(\pi)$ returns the Permutation Forest of π (i.e., the set of PETs that can generate the permutation π), Δ represents a permutation tree, d represents a derivation of a permutation tree and r represents a production rule. Efficient estimation for this model is done by using the standard Inside-Outside algorithm (Lari and Young, 1990).

At test time, the source sentence is parsed with the estimated grammar in order to find the derivation of a permutation tree with the lowest expected cost. More formally, the decoding task can be described as:

$$\hat{d} = \operatorname*{arg\,min}_{d \in \mathrm{Chart}(\mathbf{s})} \sum_{d' \in \mathrm{Chart}(\mathbf{s})} P(d') \operatorname{cost}(d, d')$$

where $P(d) = \prod_{r \in d} P(r)$ is the probability of a derivation, and $\mathrm{Chart}(\mathbf{s})$ is the space of all possible derivations of all possible permutation trees for source sentence $\mathbf{s}$. Two main modifications to this formula are made in order to make inference fast: First, Kendall τ is used as a cost function because it decomposes well,[8] which allows usage of efficient dynamic programming minimum Bayes-risk (MBR) computation (DeNero et al., 2009). Second, instead of computing the MBR derivation over the full chart, computation is done over 10,000 unbiased samples from the chart. To build the permutation lattice with this model we use the top n permutations which have the lowest expected Kendall τ cost.



(a) Linear form. (b) Minimized lattice.

Figure 3: Example permutation lattice.

5 Machine Translation with Permutation Lattices

5.1 Permutation Lattices

We call a *permutation lattice* for sentence $\mathbf{s} = \langle \mathbf{s}_1, \dots, \mathbf{s}_n \rangle$ an acyclic finite-state automaton where every path from the initial state reaches an accepting state in exactly n uniquely labeled transitions. Transitions are labeled with pairs in $\{(i, \mathbf{s}_i)_{i=1}^{n}\}$ and each path represents an arbitrary permutation of the source's n tokens.

In a permutation lattice with states Q and transitions E, every path between any two states $u, v \in Q$ has exactly the same length. Let $\text{out}^*(x)$ denote the transitive closure of $x \in Q$, that is, the set of states reachable from x. If two nodes are at all connected, $v \in \text{out}^*(u)$, then the distance between them equals $d_v - d_u$, where d_x is x's distance from the initial state. This observation allows a speed up of non-monotone translation of a permutation lattice. Namely, to precompute shortest distances, necessary to impose a distortion limit, instead of running a fully fledged all-pairs shortest path algorithm $O(|Q|^3)$ (Cormen et al., 2001), we can compute transitive closure in time $O(|Q| \times |E|)$ (Simon, 1988) followed by single-source distance in time $O(|Q| + |E|)$ (Mohri, 2002).

We produce permutation lattices by compressing the n-best outputs from the reordering models into a minimal deterministic acceptor. Unweighted determinization and minimization are performed using OpenFST (Allauzen et al., 2007). The results of this process are very compact representations that can be decoded efficiently. As an illustration, Figure 3 shows an English sentence from WMT newstest 2014 preordered for translation into German before (3a) and after minimization (3b).[9] Table 1 shows the influence of the number of predicted permutations on the lattice sizes

[8] More precisely, we use the Kendall τ distance between the permutations that are yields of the derivations.

[9] Example sentence: *The Kluser lights protect cyclists, as well as those travelling by bus and the residents of Bergle.*

for English–German. Permutation quality is measured by Kendall τ distance to the gold permutation (best-out-of-*n*).

		Lattice	
Permutations	Kendall τ	States	Transitions
Monotone	83.78	23	22
5	84.69	24	52
10	85.23	33	69
100	86.20	72	138
1000	86.75	123	233

Table 1: Permutations and lattice size (En–De).

5.2 Lattice Silver Training

While for first-best word order predictions, there are two straight-forward options for how to select training instances for the MT system, it is less clear how to do this in the case of permutation lattices. In standard preordering, the word order of the source sentence in the training set is commonly determined by reordering the source sentence to minimize the number of crossing alignment links (we denote this as $\mathbf{s}'$). Alternatively, the trained preordering model can be applied to the source side of the training set, which we call $\hat{\mathbf{s}}'_1$. There is a trade-off between both methods: While $\mathbf{s}'$ will generally produce more compact and less noisy phrase tables, it may include phrases that are not reachable by the preordering model. The predicted order $\hat{\mathbf{s}}'_1$, on the other hand, may be too constrained to reach helpful hypotheses. For lattices, one option would be to extract all possible phrases from the lattice directly. Here, we consider a simpler alternative: Instead of selecting either the gold order $\mathbf{s}'$ or the predicted order $\hat{\mathbf{s}}'_1$, we select the order $\hat{\mathbf{s}}'$ which is closest to both the lattice predictions and the gold order $\mathbf{s}'$. Since this order is a mix of the lattice predictions and the gold order, we call this training scheme lattice silver training.

Let $(\mathbf{s}, \mathbf{t})$ be a training instance consisting of a source sentence $\mathbf{s}$ and a target sentence $\mathbf{t}$ and let $\mathbf{s}'$ be the target-order source sentence obtained via the word alignments. For each training instance, we select the preordered source $\hat{\mathbf{s}}'$ as follows:

$$\hat{\mathbf{s}}' = \underset{\hat{\mathbf{s}}'_L \in \pi_k(\mathbf{s})}{\arg\max}\ \mathrm{overlap}(\hat{\mathbf{s}}'_L, \mathbf{s}')$$

where $\pi_k(\mathbf{s})$ is the set of k-best permutations predicted by the preordering model. Each $\hat{\mathbf{s}}'_L \in \pi_k(\mathbf{s})$ represents a single path through the lattice. As the cost function, we use n-gram overlap, as commonly used in string kernels (Lodhi et al., 2002):

$$\mathrm{overlap}(\hat{\mathbf{s}}'_L, \mathbf{s}') = \sum_{n=2}^{7} \left(\sum_{c \in C^n_{\mathbf{s}'}} \mathrm{count}_{\hat{\mathbf{s}}'_L}(c) \right)$$

where $C^n_{\mathbf{s}'}$ denotes all candidate n-grams of length n in $\mathbf{s}'$ and $\mathrm{count}_{\hat{\mathbf{s}}'_L}(c)$ denotes the number of occurrences of n-gram c in $\hat{\mathbf{s}}'_L$. Ties between permutations with the same overlap are broken using the permutations' scores from the preordering model.

6 Experiments

6.1 Experimental Setup

In our translation experiments, we use the following experimental setup, datasets and parameters.

Translation system Translation experiments are performed with a phrase-based machine translation system, a version of Moses (Koehn et al., 2007) with extended lattice support.[10] We use the basic Moses features and perform 15 iterations of batch MIRA (Cherry and Foster, 2012).

English–Japanese Our experiments are performed on the NTCIR-8 Patent Translation (PATMT) Task. Tuning is performed on the NTCIR-7 dev sets, and translation is evaluated on the test set from NTCIR-9. All data is tokenized (using the Moses tokenizer for English and KyTea 5 for Japanese (Neubig et al., 2011)) and filtered for sentences between 4 and 50 words. As a baseline we use a translation system with distortion limit 6 and a lexicalized reordering model (Galley and Manning, 2008). We use a 5-gram language model estimated using *lmplz* (Heafield et al., 2013) on the target side of the parallel corpus.

English–German For translation into German, we built a machine translation system based on the WMT 2016 news translation data.[11] The system is trained on all available parallel data, consisting of 4.5m sentence pairs from Europarl (Koehn, 2005), Common Crawl (Smith et al., 2013) and the News Commentary corpus. We removed all sentences longer than 80 words and tokenization and truecasing is performed using the standard Moses tokenizer and truecaser. We use a 5-gram Kneser-Ney language model, estimated using *lmplz* (Heafield

[10] Made available at `https://github.com/wilkeraziz/mosesdecoder`.

[11] `http://statmt.org/wmt16/`

et al., 2013). The language model is trained on 189m sentences from the target sides of Europarl and News Commentary, as well as the News Crawl 2007-2015 corpora. Word alignment is performed using MGIZA (*gdfa* with 6, 6, 3 and 3 iterations of IBM M1, HMM, IBM M3 and IBM M4). As a baseline we use a translation system with distortion limit 6 and a distortion-based reordering model. Tuning is performed on newstest 2014 and we evaluate on newstest 2015.

Preordering models For German, we use the neural lattice preordering model introduced in Section 4.1. The model is trained on the full parallel training data (4.5m sentences) based on the automatic word alignments used by the translation system. Source dependency trees are produced by TurboParser,[12] which was trained on the English version of HamleDT (Zeman et al., 2012) with content-head dependencies. For translation into Japanese, we train a Reordering Grammar model for 10 iterations of EM on a training set consisting of 786k sentence pairs with automatic alignments.

6.2 Translation Experiments

We report lowercased BLEU (Papineni et al., 2002) and Kendall τ calculated from the force-aligned hypothesis and reference. Statistical significance tests are performed for the translation scores using the bootstrap resampling method with p-value < 0.05 (Koehn, 2004). The standard preordering systems ("first-best" in Table 2 and 4) use an additional lexicalized reordering model (MSD), while the lattice systems use only lattice distortion. For training preordered translation models, we recreate word alignments from the original MGIZA alignments and the permutation for En–De and re-align preordered and target sentences for En–Ja using MGIZA.[13]

English–German Translation results for translation into German are shown in Table 2.

For this language pair, we found standard preordering to work poorly. This is despite the fact that the oracle order (i.e. the source words in the test set are preordered according to the word alignments) shows significant potential. A lattice packed with 1000 permutations on the other hand, performs better even when translating monotonically with a distortion limit of 0.

		Translation	Word order
	DL	BLEU	Kendall τ
Baseline	6	21.76	54.75
Oracle order	6	26.68	58.05
	0	26.41	57.92
First-best	6	21.21^{A}	53.44
Lattice (silver)	0	21.88^{B}	54.51

^{A}Stat. significant against baseline. ^{B}Stat. significant against first-best.

Table 2: Translation results English–German.

Lattice silver training To examine the utility of the lattice silver training scheme, we train systems which differ only in the way the training data is extracted. Table 3 shows that for English–German, lattice silver training is successful in bridging the gap between the preordering model and the alignment-based target word order, both for monotonic translation and when allowing the decoder to additionally reorder translations.

	Distortion limit	
	0	3
Gold training	21.44	21.60
Lattice silver training	21.88	21.88

Table 3: Lattice silver training (BLEU, En–De).

English–Japanese Results for translation into Japanese are shown in Table 4.

Discussion Although preordering with a single permutation already works well for the strict word order language Japanese, packing the word order ambiguity into a lattice allows the machine translation system to achieve even better translation monotonically than allowing a distortion of 6 and an additional lexicalized reordering model on top

		Translation	Word order
	DL	BLEU	Kendall τ
Baseline	6	29.65	44.87
Oracle order	6	34.22	56.23
	0	30.55	53.98
First-best	6	32.14^{A}	49.68
Lattice	0	32.50^{AB}	50.79

^{A}Stat. significant against baseline. ^{B}Stat. significant against first-best.

Table 4: Translation results English–Japanese.

[12] `http://cs.cmu.edu/~ark/TurboParser/`

[13] Re-aligning the sentences with MGIZA generally improves results, which implies that we are likely underestimating the results for En–De.

of a single permutation. We noticed that lexicalized reordering helped the first-best systems and hence report this stronger baseline. In principle, lexicalized reordering can also be used with 0-distortion lattice translation, and we plan to investigate this option in the future. Linguistic intuition and the empirical results presented in Section 3 suggest that compared to Japanese, German shows more word order freedom. Consequently, we assumed that a first-best preordering model would not perform well on the language pair English–German, and indeed the results in Table 2 confirm this assumption. For both language pairs, translating a lattice of predicted permutations outperforms the baselines, thus reducing the gap between translation with predicted word order and oracle word order. However, permutation lattices turn out to be the key to enabling any improvement at all for the language pair English–German in the context of preordering. This language pair can benefit from the improved interaction between word order and translation decisions. These findings go in tandem with our analysis in Section 3 (see Figures 1 and 2), particularly, the prediction of our information-theoretic word order freedom metric that it should be more difficult to determine German word order from English clues. Our main focus in this paper was on the language pairs English–German and English–Japanese. Hence, while our results provide an empirical data point for the utility of permutation lattices for free word order languages, we plan to provide further empirical support by performing experiments with a broader range of language pairs in future work.

7 Conclusion

The world's languages differ widely in how they express meaning, relying on indicators such as word order, intonation or morphological markings. Consequently, some languages exhibit stricter word order than others. Our goal in this paper was to examine the effect of word order freedom on machine translation and preordering. We provided an empirical comparison of language pairs in terms of the difficulty of predicting the target language's word order based on the source language. Our metric's predictions agree both with the intuition provided by linguistic theory and the empirical support we present in the form of translation experiments. We show that addressing uncertainty in word order predictions, and in particular doing so with permutation lattices, can be an indispensable tool for dealing with word order in machine translation. The experiments we performed in this paper confirm this previous finding and we further build on it by introducing a new method for training machine translation systems for lattice-preordered input, which we call *lattice silver training*. Finally, we found that while lattices are indeed helpful for English–Japanese, for which standard preordering already works well, they are crucial for translation into the freer word order language German.

Acknowledgements

We thank the three anonymous reviewers for their constructive comments and suggestions. This work received funding from EXPERT (EU FP7 Marie Curie ITN nr. 317471), NWO VICI grant nr. 277-89-002 (Khalil Sima'an), DatAptor project STW grant nr. 12271 and QT21 project (H2020 nr. 645452).

References

Cyril Allauzen, Michael Riley, Johan Schalkwyk, Wojciech Skut, and Mehryar Mohri. 2007. OpenFst: A general and efficient weighted finite-state transducer library. In *Proceedings of the Ninth International Conference on Implementation and Application of Automata, (CIAA 2007)*, volume 4783 of *Lecture Notes in Computer Science*, pages 11–23. Springer. `http://www.openfst.org`.

Egon Balas and Paolo Toth. 1983. Branch and bound methods for the traveling salesman problem. Technical report, Carnegie-Mellon Univ. Pittsburgh PA Management Sciences Research Group.

Yoshua Bengio, Réjean Ducharme, Pascal Vincent, and Christian Janvin. 2003. A neural probabilistic language model. *J. Mach. Learn. Res.*, 3:1137–1155, March.

Nicola Bertoldi, Richard Zens, and Marcello Federico. 2007. Speech translation by confusion network decoding. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, volume 4 of *ICASSP '07*, pages 1297–1300, Honolulu, HI, April. IEEE.

Colin Cherry and George Foster. 2012. Batch tuning strategies for statistical machine translation. In *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 427–436, Montréal, Canada, June.

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.

Thomas H. Cormen, Clifford Stein, Ronald L. Rivest, and Charles E. Leiserson. 2001. *Introduction to Algorithms*. McGraw-Hill Higher Education, 2nd edition.

Marta R. Costa-jussà and José A. R. Fonollosa. 2006. Statistical machine reordering. In *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing*, pages 70–76, Sydney, Australia, July.

Adrià de Gispert, Gonzalo Iglesias, and Bill Byrne. 2015. Fast and accurate preordering for SMT using neural networks. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1012–1017, Denver, Colorado, May–June.

Simon DeDeo, Robert X. D. Hawkins, Sara Klingenstein, and Tim Hitchcock. 2013. Bootstrap methods for the empirical study of decision-making and information flows in social systems. *Entropy*, 15(6):2246–2276.

John DeNero and Jakob Uszkoreit. 2011. Inducing sentence structure from parallel corpora for reordering. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 193–203, Edinburgh, Scotland, UK., July.

John DeNero, David Chiang, and Kevin Knight. 2009. Fast consensus decoding over translation forests. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 2 - Volume 2*, ACL '09, pages 567–575, Stroudsburg, PA, USA.

Christopher Dyer, Smaranda Muresan, and Philip Resnik. 2008. Generalizing word lattice translation. In *Proceedings of ACL-08: HLT*, pages 1012–1020, Columbus, Ohio, June.

Christopher J. Dyer. 2007. The "noisier channel": Translation from morphologically complex languages. In *Proceedings of the Second Workshop on Statistical Machine Translation*, pages 207–211, Prague, Czech Republic, June.

M. Amin Farajian, Nicola Bertoldi, and Marcello Federico. 2014. Online word alignment for online adaptive machine translation. In *Proceedings of the EACL 2014 Workshop on Humans and Computer-assisted Translation*, pages 84–92, Gothenburg, Sweden, April.

Richard Futrell, Kyle Mahowald, and Edward Gibson. 2015. Quantifying word order freedom in dependency corpora. In *Proceedings of the Third International Conference on Dependency Linguistics (Depling 2015)*, pages 91–100, Uppsala, Sweden, August. Uppsala University, Uppsala, Sweden.

Michel Galley and Christopher D. Manning. 2008. A simple and effective hierarchical phrase reordering model. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*, pages 848–856, Honolulu, Hawaii, October.

João Graça, Joana Paulo Pardal, and Luísa Coheur. 2008. Building a golden collection of parallel multi-language word alignments.

Kenneth Heafield, Ivan Pouzyrevsky, Jonathan H. Clark, and Philipp Koehn. 2013. Scalable modified Kneser-Ney language model estimation. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics*, pages 690–696, Sofia, Bulgaria, August.

Teresa Herrmann, Jochen Weiner, Jan Niehues, and Alex Waibel. 2013. Analyzing the potential of source sentence reordering in statistical machine translation. In *Proceedings of the International Workshop on Spoken Language Translation (IWSLT 2013)*.

Laura Jehl, Adrià de Gispert, Mark Hopkins, and Bill Byrne. 2014. Source-side preordering for translation using logistic regression and depth-first branch-and-bound search. In *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics*, pages 239–248, Gothenburg, Sweden, April.

Maxim Khalilov and Khalil Sima'an. 2012. Statistical translation after source reordering: Oracles, context-aware models, and empirical analysis. *Natural Language Engineering*, 18:491–519, 10.

Maxim Khalilov, José A. R. Fonollosa, and Mark Dras. 2009. Coupling hierarchical word reordering and decoding in phrase-based statistical machine translation. In *Proceedings of the Third Workshop on Syntax and Structure in Statistical Translation*, SSST '09, pages 78–86, Stroudsburg, PA, USA.

Dan Klein and Christopher D. Manning. 2001. Parsing and hypergraphs. In *Seventh International Workshop on Parsing Technologies (IWPT- 2001)*, October.

Kevin Knight and Yaser Al-Onaizan. 1998. Translation with finite-state devices. In *Proceedings of the Association for Machine Translation in the Americas*, AMTA, pages 421–437, Langhorne, PA, USA.

Philipp Koehn, Franz Josef Och, and Daniel Marcu. 2003. Statistical phrase-based translation. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology - Volume 1*, NAACL '03, pages 48–54, Stroudsburg, PA, USA.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondřej Bojar, Alexandra Constantin, and Evan Herbst. 2007. Moses: Open source toolkit for statistical machine translation. In *Proceedings of the 45th Annual Meeting of the ACL on Interactive Poster and Demonstration Sessions*, ACL '07, pages 177–180, Stroudsburg, PA, USA.

Philipp Koehn. 2004. Statistical significance tests for machine translation evaluation. In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*, pages 388–395.

Philipp Koehn. 2005. Europarl: A parallel corpus for statistical machine translation. In *Proceedings of Machine Translation Summit X*, volume 5, pages 79–86.

Vladislav Kuboň and Markéta Lopatková. 2015. Free or fixed word order: What can treebanks reveal? In Jakub Yaghob, editor, *ITAT 2015: Information Technologies Applications and Theory, Proceedings of the 15th conference ITAT 2015*, volume 1422 of *CEUR Workshop Proceedings*, pages 23–29, Praha, Czechia. Charles University in Prague, CreateSpace Independent Publishing Platform.

Shankar Kumar and William Byrne. 2003. A weighted finite state transducer implementation of the alignment template model for statistical machine translation. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology - Volume 1*, NAACL '03, pages 63–70, Stroudsburg, PA, USA.

K. Lari and S. J. Young. 1990. The estimation of stochastic context-free grammars using the inside-outside algorithm. *Computer Speech and Language*, 4:35–56.

Uri Lerner and Slav Petrov. 2013. Source-side classifier preordering for machine translation. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 513–523, Seattle, Washington, USA, October.

Huma Lodhi, Craig Saunders, John Shawe-Taylor, Nello Cristianini, and Chris Watkins. 2002. Text classification using string kernels. *J. Mach. Learn. Res.*, 2:419–444, March.

Takuya Matsuzaki, Yusuke Miyao, and Jun'ichi Tsujii. 2005. Probabilistic CFG with latent annotations. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05)*, pages 75–82, Ann Arbor, Michigan, June.

Mehryar Mohri. 2002. Semiring frameworks and algorithms for shortest-distance problems. *Journal of Automata, Languages and Combinatorics*, 7(3):321–350, January.

Graham Neubig, Yosuke Nakata, and Shinsuke Mori. 2011. Pointwise prediction for robust, adaptable japanese morphological analysis. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 529–533, Portland, Oregon, USA, June.

Graham Neubig. 2011. The Kyoto free translation task. http://www.phontron.com/kftt.

Hermann Ney. 1999. Speech translation: coupling of recognition and translation. In *IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 1, pages 517–520, Phoenix, AZ, March. IEEE.

Franz Josef Och and Hermann Ney. 2003. A systematic comparison of various statistical alignment models. *Computational Linguistics*, 29(1):19–51.

Sebastian Padó and Mirella Lapata. 2006. Optimal constituent alignment with edge covers for semantic projection. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*, pages 1161–1168, Sydney, Australia, July.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. BLEU: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting on association for computational linguistics*, pages 311–318.

Slav Petrov, Leon Barrett, Romain Thibaux, and Dan Klein. 2006. Learning accurate, compact, and interpretable tree annotation. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th Annual Meeting of the Association for Computational Linguistics*, ACL-44, pages 433–440, Stroudsburg, PA, USA.

Detlef Prescher. 2005. Inducing head-driven pcfgs with latent heads: Refining a tree-bank grammar for parsing. In *In ECML05*.

K. Simon. 1988. An improved algorithm for transitive closure on acyclic digraphs. *Theor. Comput. Sci.*, 58(1-3):325–346, June.

Jason R. Smith, Herve Saint-Amand, Magdalena Plamada, Philipp Koehn, Chris Callison-Burch, and Adam Lopez. 2013. Dirt cheap web-scale parallel text from the common crawl. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1374–1383, Sofia, Bulgaria, August.

Miloš Stanojević and Khalil Sima'an. 2015. Reordering grammar induction. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 44–54, Lisbon, Portugal, September.

Jörg Tiedemann. 2012. Parallel data, tools and interfaces in opus. In Nicoletta Calzolari (Conference Chair), Khalid Choukri, Thierry Declerck, Mehmet Ugur Dogan, Bente Maegaard, Joseph Mariani, Jan Odijk, and Stelios Piperidis, editors, *Proceedings of the Eight International Conference on Language Resources and Evaluation (LREC'12)*, Istanbul, Turkey, may.

Roy Tromble and Jason Eisner. 2009. Learning linear ordering problems for better translation. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*, pages 1007–1016, Singapore, August.

Edo S van der Poort, Marek Libura, Gerard Sierksma, and Jack A.A van der Veen. 1999. Solving the k-best traveling salesman problem. *Computers & Operations Research*, 26(4):409 – 425.

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

Jia Xu, Evgeny Matusov, Richard Zens, and Hermann Ney. 2005. Integrated chinese word segmentation in statistical machine translation. In *International Workshop on Spoken Language Translation*, Pittsburgh.

Daniel Zeman, David Mareček, Martin Popel, Loganathan Ramasamy, Jan Štěpánek, Zdeněk Žabokrtský, and Jan Hajič. 2012. Hamledt: To parse or not to parse? In *Proceedings of the Eight International Conference on Language Resources and Evaluation (LREC'12)*, Istanbul, Turkey, may. European Language Resources Association (ELRA).

Hao Zhang and Daniel Gildea. 2007. Factorization of synchronous context-free grammars in linear time. In *NAACL Workshop on Syntax and Structure in Statistical Translation (SSST)*, pages 25–32.

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 the NAACL-HLT 2007/AMTA Workshop on Syntax and Structure in Statistical Translation*, SSST '07, pages 1–8, Stroudsburg, PA, USA.

Daniel Zwillinger and Stephen Kokoska. 1999. *CRC Standard Probability and Statistics Tables and Formulae*. CRC Press.