Evaluating Word Order Recursively over Permutation-Forests

Miloš Stanojević and **Khalil Sima'an**
Institute for Logic, Language and Computation
University of Amsterdam
Science Park 107, 1098 XG Amsterdam, The Netherlands
{m.stanojevic,k.simaan}@uva.nl

Abstract

Automatically evaluating word order of MT system output at the sentence-level is challenging. At the sentence-level, ngram counts are rather sparse which makes it difficult to measure word order quality effectively using lexicalized units. Recent approaches abstract away from lexicalization by assigning a score to the *permutation* representing how word positions in system output move around relative to a reference translation. Metrics over permutations exist (e.g., Kendal tau or Spearman Rho) and have been shown to be useful in earlier work. However, none of the existing metrics over permutations groups word positions recursively into larger phrase-like blocks, which makes it difficult to account for long-distance reordering phenomena. In this paper we explore novel metrics computed over *Permutation Forests (PEFs)*, packed charts of Permutation Trees (PETs), which are tree decompositions of a permutation into primitive ordering units. We empirically compare PEFs metric against five known reordering metrics on WMT13 data for ten language pairs. The PEFs metric shows better correlation with human ranking than the other metrics almost on all language pairs. None of the other metrics exhibits as stable behavior across language pairs.

1 Introduction

Evaluating word order (also reordering) in MT is one of the main ingredients in automatic MT evaluation, e.g., (Papineni et al., 2002; Denkowski and Lavie, 2011). To monitor progress on evaluating reordering, recent work explores dedicated reordering evaluation metrics, cf. (Birch and Osborne, 2011; Isozaki et al., 2010; Talbot et al., 2011). Existing work computes the correlation between the ranking of the outputs of different systems by an evaluation metric to human ranking, on e.g., the WMT evaluation data.

For evaluating reordering, it is necessary to word align system output with the corresponding reference translation. For convenience, a 1:1 alignment (a permutation) is induced between the words on both sides (Birch and Osborne, 2011), possibly leaving words unaligned on either side. Existing work then concentrates on defining measures of reordering over permutations, cf. (Lapata, 2006; Birch and Osborne, 2011; Isozaki et al., 2010; Talbot et al., 2011). Popular metrics over permutations are: Kendall's tau, Spearman, Hamming distance, Ulam and Fuzzy score. These metrics treat a permutation as a flat sequence of integers or blocks, disregarding the possibility of hierarchical grouping into phrase-like units, making it difficult to measure long-range order divergence. Next we will show by example that permutations also contain latent atomic units that govern the recursive reordering of phrase-like units. Accounting for these latent reorderings could actually be far simpler than the flat view of a permutation.

Isozaki et al. (2010) argue that the conventional metrics cannot measure well the long distance reordering between an English reference sentence "A because B" and a Japanese-English hypothesis translation "B because A", where A and B are blocks of any length with internal monotonic alignments. In this paper we explore the idea of factorizing permutations into permutation-trees (PETs) (Gildea et al., 2006) and defining new

Proceedings of SSST-8, Eighth Workshop on Syntax, Semantics and Structure in Statistical Translation, pages 138–147,
October 25, 2014, Doha, Qatar.

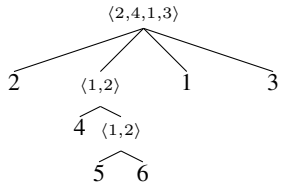
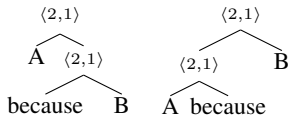


Figure 1: A permutation tree for $\langle 2, 4, 5, 6, 1, 3 \rangle$

tree-based reordering metrics which aims at dealing with this type of long range reorderings. For the Isozaki et al. (2010) Japanese-English example, there are two PETs (when leaving A and B as encapsulated blocks):



Our PET-based metrics interpolate the scores over the two inversion operators $\langle 2, 1 \rangle$ with the internal scores for A and B, incorporating a weight for subtree height. If both A and B are large blocks, internally monotonically (also known as straight) aligned, then our measure will not count every single reordering of a word in A or B, but will consider this case as block reordering. From a PET perspective, the distance of the reordering is far smaller than when looking at a flat permutation. But does this hierarchical view of reordering cohere better with human judgement than string-based metrics?

The example above also shows that a permutation may factorize into different PETs, each corresponding to a different segmentation of a sentence pair into phrase-pairs. In this paper we introduce *permutation forests (PEFs)*; a PEF is a hypergraph that compactly packs the set of PETs that factorize a permutation.

There is yet a more profoud reasoning behind PETs than only accounting for long-range reorderings. The example in Figure 1 gives the flavor of PETs. Observe how every internal node in this PET dominates a subtree whose fringe[1] is itself a permutation over an *integer sub-range* of the original permutation. Every node is decorated with a permutation over the child positions (called operator). For example $\langle 4, 5, 6 \rangle$ constitutes a contiguous range of integers (corresponding to a phrase pair), and hence will be grouped into a subtree; which in turn can be internally re-grouped into a binary branching subtree. Every node in a PET is *minimum branching*, i.e., the permutation factorizes into a minimum number of adjacent permutations over integer sub-ranges (Albert and Atkinson, 2005). The node operators in a PET are known to be the atomic building blocks of all permutations (called primal permutations). Because these are building atomic units of reordering, it makes sense to want to measure reordering as a function of the individual cost of these operators. In this work we propose to compute new reordering measures that aggregate over the individual node-permutations in these PETs.

While PETs where exploited rather recently for extracting features used in the BEER metric *system description* (Stanojević and Sima'an, 2014) in the official WMT 2014 competition, this work is the first to propose integral *recursive* metrics over PETs and PEFs solely for measuring *reordering* (as opposed to individual non-recursive features in a full metric that measures at the same time both fluency and adequacy). We empirically show that a PEF-based evaluation measure correlates better with human rankings than the string-based measures on *eight* of the ten language pairs in WMT13 data. For the 9^{th} language pair it is close to best, and for the 10^{th} (English-Czech) we find a likely explanation in the *Findings of the 2013 WMT* (Bojar et al., 2013). Crucially, the PEF-based measure shows more stable ranking across language pairs than any of the other measures. The metric is available online as free software[2].

2 Measures on permutations: Baselines

In (Birch and Osborne, 2010; Birch and Osborne, 2011) Kendall's tau and Hamming distance are combined with unigram BLEU (BLEU-1) leading to LRscore showing better correlation with human judgment than BLEU-4. Birch et al. (2010) additionally tests Ulam distance (longest common subsequence – LCS – normalized by the permutation length) and the square root of Kendall's tau. Isozaki et al. (2010) presents a similar approach to (Birch and Osborne, 2011) additionally testing Spearman rho as a distance measure. Talbot et al. (2011) extracts a reordering measure from METEOR (Denkowski and Lavie, 2011) dubbed *Fuzzy Reordering Score* and evaluates it on MT reordering quality.

[1] Ordered sequence of leaf nodes.

[2] https://github.com/stanojevic/beer

For an evaluation metric we need a function which would have the standard behaviour of evaluation metrics - the higher the score the better. Bellow we define the *baseline metrics* that were used in our experiments.

Baselines A permutation over $[1..n]$ (subrange of the positive integers where $n > 1$) is a bijective function from $[1..n]$ to itself. To represent permutations we will use angle brackets as in $\langle 2, 4, 3, 1\rangle$. Given a permutation π over $[1..n]$, the notation π_i $(1 \leq i \leq n)$ stands for the integer in the i^{th} position in π; $\pi(i)$ stands for the index of the position in π where integer i appears; and π_i^j stands for the (contiguous) sub-sequence of integers $\pi_i, \ldots \pi_j$.

The definitions of five commonly used metrics over permutations are shown in Figure 2. In these definitions, we use LCS to stand for Longest Common Subsequence, and Kronecker $\delta[a]$ which is 1 if $(a == true)$ else zero, and $\mathcal{A}_1^n = \langle 1, \cdots, n\rangle$ which is the identity permutation over $[1..n]$. We note that all existing metrics

$$kendall(\pi) = \frac{\sum_{i=1}^{n-1}\sum_{j=i+1}^{n}\delta[\pi(i) < \pi(j)]}{(n^2-n)/2}$$

$$hamming(\pi) = \frac{\sum_{i=1}^{n}\delta[\pi_i == i]}{n}$$

$$spearman(\pi) = 1 - \frac{3\sum_{i=1}^{n}(\pi_i - i)^2}{n(n^2-1)}$$

$$ulam(\pi) = \frac{LCS(\pi, \mathcal{A}_1^n) - 1}{n-1}$$

$$fuzzy(\pi) = 1 - \frac{c-1}{n-1}$$

where c is # of monotone sub-permutations

Figure 2: Five commonly used metrics over permutations

are defined directly over flat string-level permutations. In the next section we present an alternative view of permutations are compositional, recursive tree structures.

3 Measures on Permutation Forests

Existing work, e.g., (Gildea et al., 2006), shows how to **factorize** any permutation π over $[1..n]$ into a canonical permutation tree (PET). Here we will summarize the relevant aspects and extend PETs to permutation forests (PEFs).

A non-empty sub-sequence π_i^j of a permutation π is *isomorphic* with a permutation over $[1..(j - i + 1)]$ iff the set $\{\pi_i, \ldots, \pi_j\}$ is a *contiguous range* of positive integers. We will use the term a **sub-permutation** of π to refer to a subsequence of π that is isomorphic with a permutation. Note that not every subsequence of a permutation π is necessarily isomorphic with a permutation, e.g., the subsequence $\langle 3, 5\rangle$ of $\langle 1, 2, 3, 5, 4\rangle$ is not a sub-permutation. One sub-permutation π_1 of π is **smaller** than another sub-permutation π_2 of π iff every integer in π_1 is smaller than all integers in π_2. In this sense we can put a full order on *non-overlapping* sub-permutations of π and rank them from the smallest to the largest.

For every permutation π there is a *minimum number* of adjacent sub-permutations it can be factorized into (see e.g., (Gildea et al., 2006)). We will call this minimum number the **arity** of π and denote it with $\mathbf{a}(\pi)$ (or simply a when π is understood from the context). For example, the arity of $\pi = \langle 5, 7, 4, 6, 3, 1, 2\rangle$ is $a = 2$ because it can be split into a minimum of two sub-permutations (Figure 3), e.g. $\langle 5, 7, 4, 6, 3\rangle$ and $\langle 1, 2\rangle$ (but alternatively also $\langle 5, 7, 4, 6\rangle$ and $\langle 3, 1, 2\rangle$). In contrast, $\pi = \langle 2, 4, 1, 3\rangle$ (also known as the Wu (1997) permutation) cannot be split into less than four sub-permutations, i.e., $a = 4$. Factorization can be applied recursively to the sub-permutations of π, resulting in a tree structure (see Figure 3) called a permutation tree (PET) (Gildea et al., 2006; Zhang and Gildea, 2007; Maillette de Buy Wenniger and Sima'an, 2011).

Some permutations factorize into multiple alternative PETs. For $\pi = \langle 4, 3, 2, 1\rangle$ there are five PETs shown in Figure 3. The alternative PETs can be packed into an $O(n^2)$ permutation forest (PEF). For many computational purposes, a single *canonical PET* is sufficient, cf. (Gildea et al., 2006). However, while different PETs of π exhibit the same reordering pattern, their different binary branching structures might indicate important differences as we show in our experiments.

A **permutation forest** (akin to a parse forest) $\mathcal{F}$ for π (over $[1..n]$) is a data structure consisting of a subset of $\{[[i, j, \mathcal{I}_i^j, O_i^j]] \mid 0 \leq i \leq j \leq n\}$, where $\mathcal{I}_i^j$ is a (possibly empty) set of *inferences* (sets of split points) for π_{i+1}^j and O_i^j is an operator shared by all inferences of π_{i+1}^j. If π_{i+1}^j is a sub-permutation and it has arity $a \leq (j - (i +$

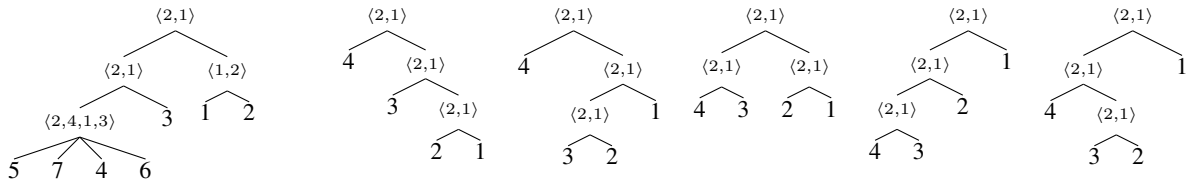


Figure 3: A PET for $\pi = \langle 5, 7, 4, 6, 3, 1, 2\rangle$. And five different PETs for $\pi = \langle 4, 3, 2, 1\rangle$.

1)), then each inference consists of a $a-1$-tuple $[l_1, \ldots, l_{a-1}]$, where for each $1 \leq x \leq (a-1)$, l_x is a "split point" which is given by the index of the last integer in the x^{th} sub-permutation in π. The permutation of the a sub-permutations ("children" of π^j_{i+1}) is stored in O^j_i and it is the same for all inferences of that span (Zhang et al., 2008).

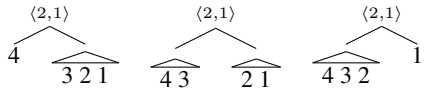


Figure 4: The factorizations of $\pi = \langle 4, 3, 2, 1\rangle$.

Let us exemplify the inferences on $\pi = \langle 4, 3, 2, 1\rangle$ (see Figure 4) which factorizes into pairs of sub-permutations ($a = 2$): a split point can be at positions with index $l_1 \in \{1, 2, 3\}$. Each of these split points (factorizations) of π will be represented as an *inference* for the *same root node* which covers the whole of π (placed in entry $[0, 4]$); the operator of the inference here consists of the permutation $\langle 2, 1\rangle$ (swapping the two ranges covered by the children sub-permutations) and inference consists of $a-1$ indexes $l_1, \ldots, l_{a-1}$ signifying the split points of π into sub-permutations: since $a = 2$ for π, then a single index $l_1 \in \{1, 2, 3\}$ is stored with every inference. For the factorization $((4, 3), (2, 1))$ the index $l_1 = 2$ signifying that the second position is a split point into $\langle 4, 3\rangle$ (stored in entry $[0, 2]$) and $\langle 2, 1\rangle$ (stored in entry $[2, 4]$). For the other factorizations of π similar inferences are stored in the permutation forest.

Figure 5 shows a simple top-down factorization algorithm which starts out by computing the arity a using function $\mathbf{a}(\pi)$. If $a = 1$, a single leaf node is stored with an empty set of inferences. If $a > 1$ then the algorithm computes all possible factorizations of π into a sub-permutations (a sequence of $a-1$ split points) and stores their inferences together as $\mathcal{I}^j_i$ and their operator O^j_i associated with a node in entry $[[i, j, \mathcal{I}^j_i, O^j_i]]$. Subsequently, the algorithm applies recursively to each sub-permutation. Efficiency is a topic beyond the scope of this paper, but this naive algorithm has worst case time complexity $O(n^3)$, and when computing only a single canonical PET this can be $O(n)$ (see e.g., (Zhang and Gildea, 2007)).

Function $PEF(i, j, \pi, \mathcal{F})$;
`# Args: sub-perm. π over [i..j] and forest F`
Output: Parse-Forest $\mathcal{F}(\pi)$ for π;

begin
if ($[[i, j, \star]] \in \mathcal{F}$) then return $\mathcal{F}$; `#memoization`
$a := \mathbf{a}(\pi)$;
if $a = 1$ return $\mathcal{F} := \mathcal{F} \cup \{[[i, j, \emptyset]]\}$;
For each set of split points $\{l_1, \ldots, l_{a-1}\}$ do
  $O^j_i := RankListOf(\pi^{l_1}_{(l_0+1)}, \pi^{l_2}_{(l_1+1)}, \ldots, \pi^{l_a}_{(l_{a-1}+1)})$;
  $\mathcal{I}^j_i := \mathcal{I}^j_i \cup [l_1, \ldots, l_{a-1}]$;
  For each $\pi_v \in \{\pi^{l_1}_{l_0+1}, \pi^{l_2}_{(l_1+1)}, \ldots, \pi^{l_a}_{(l_{a-1}+1)}\}$ do
    $\mathcal{F} := \mathcal{F} \cup PermForest(\pi_v)$;
$\mathcal{F} := \mathcal{F} \cup \{[[i, j, \mathcal{I}^j_i, O^j_i]]\}$;
Return $\mathcal{F}$;
end;

Figure 5: Pseudo-code of permutation-forest factorization algorithm. Function $\mathbf{a}(\pi)$ returns the arity of π. Function $RankListOf(r_1, \ldots, r_m)$ returns the list of rank positions (i.e., a permutation) of sub-permutations $r_1, \ldots, r_m$ after sorting them smallest first. The top-level call to this algorithm uses π, $i = 0$, $j = n$ and $\mathcal{F} = \emptyset$.

Our measure ($PEFscore$) uses a function $opScore(p)$ which assigns a score to a given operator, which can be instantiated to any of the existing scoring measures listed in Section 2, but in this case we opted for a very simple function which gives score 1 to monotone permutation and score 0 to any other permutation.

Given an inference $l \in \mathcal{I}^j_i$ where $l = [l_1, \ldots, l_{a-1}]$, we will use the notation l_x to refer to split point l_x in l where $1 \leq x \leq (a-1)$, with the convenient boundary assumption that $l_0 = i$ and $l_a = j$.

$$PEFscore(\pi) \;=\; \phi_{node}(0, n, PEF(\pi))$$

$$\phi_{node}(i, j, \mathcal{F}) \;=\; \begin{cases} \text{if } (\mathcal{I}_i^j == \emptyset) \text{ then } 1 \\ \text{else if } (\mathbf{a}(\pi_{i+1}^j) = j - i) \text{ then } opScore(O_i^j) \\ \text{else } \beta \times opScore(O_i^j) + (1-\beta) \times \underbrace{\frac{\sum_{l \in \mathcal{I}_i^j} \phi_{inf}(l, \mathcal{F}, \mathbf{a}(\pi_{i+1}^j))}{|\mathcal{I}_i^j|}}_{\text{Avg. inference score over } \mathcal{I}_i^j} \end{cases}$$

$$\phi_{inf}(l, \mathcal{F}, a) \;=\; \underbrace{\frac{\sum_{x=1}^{a} \delta[l_x - l_{x-1} > 1] \times \phi_{node}(l_{(x-1)}, l_x, \mathcal{F})}{\sum_{x=1}^{a} \delta[l_x - l_{(x-1)} > 1]}}_{\text{Avg. score for non-terminal children}}$$

$$opScore(p) \;=\; \begin{cases} \text{if } (p == \langle 1, 2 \rangle) \text{ then } 1 \\ \text{else } 0 \end{cases}$$

Figure 6: The PEF Score

The PEF-score, $PEFscore(\pi)$ in Figure 6, computes a score for the single root node $[[0, n, \mathcal{I}_0^n, O_0^n]])$ in the permutation forest. This score is the average inference score ϕ_{inf} over all inferences of this node. The score of an inference ϕ_{inf} interpolates (β) between the $opScore$ of the operator in the current span and $(1 - \beta)$ the scores of each child node. The interpolation parameter β can be tuned on a development set.

The PET-score (single PET) is a simplification of the PEF-score where the summation over all inferences of a node $\sum_{l \in \mathcal{I}_i^j}$ in ϕ_{node} is replaced by "Select a canonical $l \in \mathcal{I}_i^j$".

4 Experimental setting

Data The data that was used for experiments are human rankings of translations from WMT13 (Bojar et al., 2013). The data covers 10 language pairs with a diverse set of systems used for translation. Each human evaluator was presented with 5 different translations, source sentence and a reference translation and asked to rank system translations by their quality (ties were allowed).[3]

Meta-evaluation The standard way for doing meta-evaluation on the sentence level is with Kendall's tau correlation coefficient (Callison-Burch et al., 2012) computed on the number of times an evaluation metric and a human evaluator agree (and disagree) on the rankings of pairs of translations. We extract pairs of translations from human evaluated data and compute their scores with all metrics. If the ranking assigned by a metric is the same as the ranking assigned by a human evaluator then that pair is considered concordant, otherwise it is a discordant pair. All pairs which have the same score by the metric or are judged as ties by human evaluators are not used in meta-evaluation. The formula that was used for computing Kendall's tau correlation coefficient is shown in Equation 1. Note that the formula for Kendall tau rank correlation coefficient that is used in meta-evaluation is different from the Kendall tau similarity function used for evaluating permutations. The values that it returns are in the range $[-1, 1]$, where -1 means that order is always opposite from the human judgment while the value 1 means that metric ranks the system translations in the same way as humans do.

$$\tau = \frac{\#concordant\ pairs - \#discordant\ pairs}{\#concordant\ pairs + \#discordant\ pairs} \qquad (1)$$

Evaluating reordering Since system translations do not differ only in the word order but also in lexical choice, we follow Birch and Osborne (2010) and interpolate the score given by each reordering metric with the same lexical score. For lexical scoring we use unigram BLEU. The parameter that balances the weights for these two metrics α is chosen to be 0.5 so it would not underestimate the lexical differences between translations ($\alpha \ll 0.5$) but also would not turn the whole metric into unigram BLEU ($\alpha \gg 0.5$). The equation

[3] We would like to extend our work also to English-Japanese but we do not have access to such data at the moment. In any case, the WMT13 data is the largest publicly available data of this kind.

for this interpolation is shown in Equation 2.[4]

$$FullMetric(ref, sys) = \alpha \ lexical(ref, sys) + (1 - \alpha) \times bp(|ref|, |\pi|) \times ordering(\pi) \quad (2)$$

Where $\pi(ref, sys)$ is the permutation representing the word alignment from sys to ref. The effect of α on the German-English evaluation is visible on Figure 7. The PET and PEF measures have an extra parameter β that gives importance to the long distance errors that also needs to be tuned. On Figure 8 we can see the effect of β on German-English for $\alpha = 0.5$. For all language pairs for $\beta = 0.6$ both PETs and PEFs get good results so we picked that as value for β in our experiments.

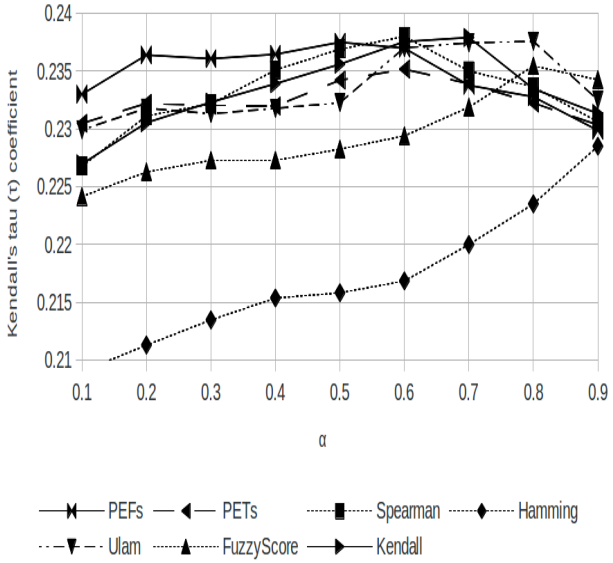


Figure 7: Effect of α on German-English evaluation for $\beta = 0.6$

Choice of word alignments The issue we did not discuss so far is how to find a permutation from system and reference translations. One way is to first get alignments between the source sentence and the system translation (from a decoder or by automatically aligning sentences), and also alignments between the source sentence and the reference translation (manually or automatically aligned). Subsequently we must make those alignments 1-to-1 and merge them into a permutation. That is the approach that was followed in previous work (Birch and Osborne, 2011; Talbot et al., 2011). Alternatively, we may align system and reference translations directly. One of the simplest ways to do that is by finding exact matches between words and bigrams between system and reference translation as done in (Isozaki et al., 2010). The way we align system and reference translations is by using the aligner supplied with METEOR (Denkowski and Lavie, 2011) for finding 1-to-1 alignments which are later converted to a permutation. The advantage of this method is that it can do non-exact matching by stemming or using additional sources for semantic similarity such as WordNets and paraphrase tables. Since we will not have a perfect permutation as input, because many words in the reference or system translations might not be aligned, we introduce a brevity penalty ($bp(\cdot, \cdot)$ in Equation 2) for the ordering component as in (Isozaki et al., 2010). The brevity penalty is the same as in BLEU with the small difference that instead of taking the length of system and reference translation as its parameters, it takes the length of the system permutation and the length of the reference.

[4]Note that for reordering evaluation it does not make sense to tune α because that would blur the individual contributions of reordering and adequacy during meta evaluation, which is confirmed by Figure 7 showing that $\alpha \gg 0.5$ leads to similar performance for all metrics.

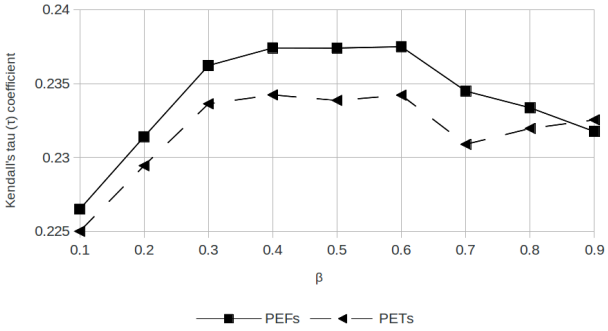


Figure 8: Effect of β on German-English evaluation for $\alpha = 0.5$

5 Empirical results

The results are shown in Table 1 and Table 2. These scores could be much higher if we used some more sophisticated measure than unigram BLEU for the lexical part (for example recall is very useful in evaluation of the system translations (Lavie et al., 2004)). However, this is not the issue here since our goal is merely to compare different ways to evaluate word order. All metrics that we tested have the same lexical component, get the same permutation as their input and have the same value for α.

	English-Czech	English-Spanish	English-German	English-Russian	English-French
Kendall	**0.16**	0.170	0.183	0.193	0.218
Spearman	0.157	0.170	0.181	0.192	0.215
Hamming	0.150	0.163	0.168	0.187	0.196
FuzzyScore	0.155	0.166	0.178	0.189	0.215
Ulam	0.159	0.170	0.181	0.189	**0.221**
PEFs	0.156	**0.173**	**0.185**	**0.196**	0.219
PETs	0.157	0.165	0.182	0.195	0.216

Table 1: Sentence level Kendall tau scores for translation out of English with $\alpha = 0.5$ and $\beta = 0.6$

	Czech-English	Spanish-English	German-English	Russian-English	French-English
Kendall	0.196	**0.265**	0.235	0.173	0.223
Spearman	0.199	**0.265**	0.236	0.173	0.222
Hamming	0.172	0.239	0.215	0.157	0.206
FuzzyScore	0.184	0.263	0.228	0.169	0.216
Ulam	0.188	0.264	0.232	0.171	0.221
PEFs	**0.201**	**0.265**	**0.237**	**0.181**	**0.228**
PETs	0.200	0.264	0.234	0.174	0.221

Table 2: Sentence level Kendall tau scores for translation into English with $\alpha = 0.5$ and $\beta = 0.6$

5.1 Does hierarchical structure improve evaluation?

The results in Tables 1, 2 and 3 suggest that the PEFscore which uses hierarchy over permutations outperforms the string based permutation metrics in the majority of the language pairs. The main exception is the English-Czech language pair in which both PETs and PEFs based metric do not give good results compared to some other metrics. For discussion about English-Czech look at the section 6.1.

5.2 Do PEFs help over one canonical PET?

From Figures 9 and 10 it is clear that using all permutation trees instead of only canonical ones makes the metric more stable in all language pairs. Not only that it makes results more stable but it also improves them in all cases except in English-Czech where both PETs and PEFs perform badly. The main reason why PEFs outperform PETs is that they encode all possible phrase segmentations of monotone and inverted sub-permutations. By giving the score that considers all segmentations, PEFs also include the right segmentation (the one perceived by human evaluators as the right segmentation), while PETs get the right segmentation only if the right segmentation is the canonical one.

metric	avg rank	avg Kendall
PEFs	1.6	0.2041
Kendall	2.65	0.2016
Spearman	3.4	0.201
PETs	3.55	0.2008
Ulam	4	0.1996
FuzzyScore	5.8	0.1963
Hamming	7	0.1853

Table 3: Average ranks and average Kendall scores for each tested metrics over all language pairs

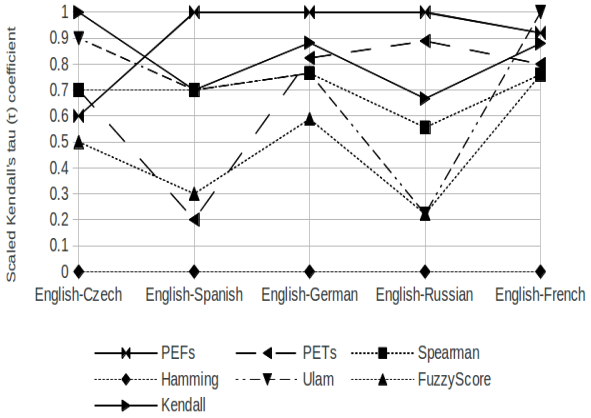


Figure 9: Plot of scaled Kendall tau correlation for translation from English

5.3 Is improvement consistent over language pairs?

Table 3 shows average rank (metric's position after sorting all metrics by their correlation for each language pair) and average Kendall tau correlation coefficient over the ten language pairs. The table shows clearly that the PEFs metric outperforms all other metrics. To make it more visible how metrics perform on the different language pairs, Figures 9 and 10 show Kendall tau correlation coefficient scaled between the best scoring metric for the given language (in most cases PEFs) and

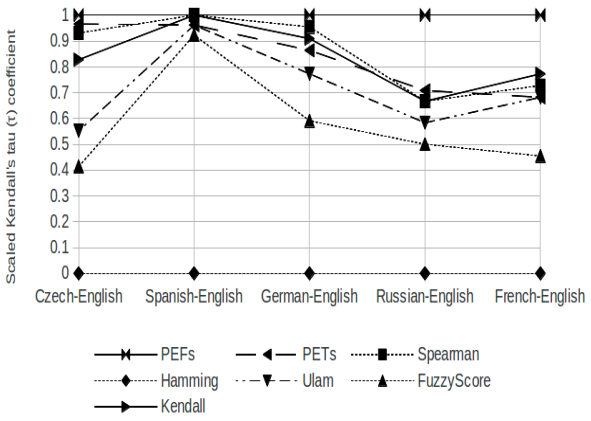


Figure 10: Plot of scaled Kendall tau correlation for translation into English

the worst scoring metric (in all cases Hamming score). We can see that, except in English-Czech, PEFs are consistently the best or second best (only in English-French) metric in all language pairs. PETs are not stable and do not give equally good results in all language pairs. Hamming distance is without exception the worst metric for evaluation since it is very strict about positioning of the words (it does not take relative ordering between words into account). Kendall tau is the only string based metric that gives relatively good scores in all language pairs and in one (English-Czech) it is the best scoring one.

6 Further experiments and analysis

So far we have shown that PEFs outperform the existing metrics over the majority of language pairs. There are two pending issues to discuss. Why is English-Czech seemingly so difficult? And does preferring inversion over non-binary branching correlate better with human judgement.

6.1 The results on English-Czech

The English-Czech language pair turned out to be the hardest one to evaluate for all metrics. All metrics that were used in the meta-evaluation that we conducted give much lower Kendall tau correlation coefficient compared to the other language pairs. The experiments conducted by other researchers on the same dataset (Macháček and Bojar, 2013), using full evaluation metrics, also get far lower Kendall tau correlation coefficient for English-Czech than for other language pairs. In the description of WMT13 data that we used (Bojar et al., 2013), it is shown that annotator-agreement for English-Czech is a few times lower than for other languages. English-Russian, which is linguistically similar to English-Czech, does not show low numbers in these categories, and is one of the language pairs where our metrics perform the best. The alignment ratio is equally high between English-Czech and English-Russian (but that does not rule out the possibility that the alignments are of different quality). One seemingly unlikely explanation is that English-Czech might be a harder task in general, and might require a more sophisticated measure. However, the more plausible explanation is that the WMT13 data for English-Czech is not of the same quality as other language pairs. It could be that data filtering, for example by taking only judgments for which many evaluators agree, could give more trustworthy results.

6.2 Is inversion preferred over non-binary branching?

Since our original version of the scoring function for PETs and PEFs on the operator level does not discriminate between kinds of non-monotone operators (all non-monotone get zero as a score) we also tested whether discriminating between inversion (binary) and non-binary operators make any difference.

	English-Czech	English-Spanish	English-German	English-Russian	English-French
PEFs $\gamma = 0.0$	0.156	0.173	**0.185**	**0.196**	**0.219**
PEFs $\gamma = 0.5$	0.157	**0.175**	0.183	0.195	**0.219**
PETs $\gamma = 0.0$	0.157	0.165	0.182	0.195	0.216
PETs $\gamma = 0.5$	**0.158**	0.165	0.183	0.195	0.217

Table 4: Sentence level Kendall tau score for translation out of English different γ with $\alpha = 0.5$ and $\beta = 0.6$

Intuitively, we might expect that inverted binary operators are preferred by human evaluators over non-binary ones. So instead of assigning zero as a score to inverted nodes we give them 0.5, while for non-binary nodes we remain with zero. The experiments with the inverted operator scored with 0.5 (i.e., $\gamma = 0.5$) are shown in Tables 4 and 5. The results show that there is no clear improvement by distinguishing between the two kinds of

	Czech-English	Spanish-English	German-English	Russian-English	French-English
PEFs $\gamma = 0.0$	0.201	**0.265**	**0.237**	**0.181**	**0.228**
PEFs $\gamma = 0.5$	0.201	0.264	0.235	0.179	0.227
PETs $\gamma = 0.0$	0.200	0.264	0.234	0.174	0.221
PETs $\gamma = 0.5$	**0.202**	0.263	0.235	0.176	0.224

Table 5: Sentence level Kendall tau score for translation into English for different γ with $\alpha = 0.5$ and $\beta = 0.6$

non-monotone operators on the nodes.

7 Conclusions

Representing order differences as compact permutation forests provides a good basis for developing evaluation measures of word order differences. These hierarchical representations of permutations bring together two crucial elements (1) grouping words into blocks, and (2) factorizing reordering phenomena recursively over these groupings. Earlier work on MT evaluation metrics has often stressed the importance of the first ingredient (grouping into blocks) but employed it merely in a flat (non-recursive) fashion. In this work we presented novel metrics based on permutation trees and forests (the PETscore and PEFscore) where the second ingredient (factorizing reordering phenomena recursively) plays a major role. Permutation forests compactly represent all possible block groupings for a given permutation, whereas permutation trees select a single canonical grouping. Our experiments with WMT13 data show that our PEFscore metric outperforms the existing string-based metrics on the large majority of language pairs, and in the minority of cases where it is not ranked first, it ranks high. Crucially, the PEFscore is by far the most stable reordering score over ten language pairs, and works well also for language pairs with long range reordering phenomena (English-German, German-English, English-Russian and Russian-English).

Acknowledgments

This work is supported by STW grant nr. 12271 and NWO VICI grant nr. 277-89-002. We thank TAUS and the other DatAptor project User Board members. We also thank Ivan Titov for helpful comments on the ideas presented in this paper.

References

Michael H. Albert and Mike D. Atkinson. 2005. Simple permutations and pattern restricted permutations. *Discrete Mathematics*, 300(1-3):1–15.

Alexandra Birch and Miles Osborne. 2010. LRscore for Evaluating Lexical and Reordering Quality in MT. In *Proceedings of the Joint Fifth Workshop on Statistical Machine Translation and MetricsMATR*, pages 327–332, Uppsala, Sweden, July. Association for Computational Linguistics.

Alexandra Birch and Miles Osborne. 2011. Reordering Metrics for MT. In *Proceedings of the Association for Computational Linguistics*, Portland, Oregon, USA. Association for Computational Linguistics.

Alexandra Birch, Miles Osborne, and Phil Blunsom. 2010. Metrics for MT evaluation: evaluating reordering. *Machine Translation*, pages 1–12.

Ondřej Bojar, Christian Buck, Chris Callison-Burch, Christian Federmann, Barry Haddow, Philipp Koehn, Christof Monz, Matt Post, Radu Soricut, and Lucia Specia. 2013. Findings of the 2013 Workshop on Statistical Machine Translation. In *Proceedings of the Eighth Workshop on Statistical Machine Translation*, pages 1–44, Sofia, Bulgaria, August. Association for Computational Linguistics.

Chris Callison-Burch, Philipp Koehn, Christof Monz, Matt Post, Radu Soricut, and Lucia Specia. 2012. Findings of the 2012 Workshop on Statistical Machine Translation. In *Proceedings of the Seventh Workshop on Statistical Machine Translation*, pages 10–51, Montréal, Canada, June. Association for Computational Linguistics.

Michael Denkowski and Alon Lavie. 2011. Meteor 1.3: Automatic Metric for Reliable Optimization and Evaluation of Machine Translation Systems. In *Proceedings of the EMNLP 2011 Workshop on Statistical Machine Translation*.

Daniel Gildea, Giorgio Satta, and Hao Zhang. 2006. Factoring Synchronous Grammars by Sorting. In *ACL*.

Hideki Isozaki, Tsutomu Hirao, Kevin Duh, Katsuhito Sudoh, and Hajime Tsukada. 2010. Automatic evaluation of translation quality for distant language pairs. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, EMNLP '10, pages 944–952, Stroudsburg, PA, USA. Association for Computational Linguistics.

Mirella Lapata. 2006. Automatic Evaluation of Information Ordering: Kendall's Tau. *Computational Linguistics*, 32(4):471–484.

Alon Lavie, Kenji Sagae, and Shyamsundar Jayaraman. 2004. The significance of recall in automatic metrics for MT evaluation. In *Proceedings of the Sixth Conference of the Association for Machine Translation in the Americas*.

Matouš Macháček and Ondřej Bojar. 2013. Results of the WMT13 Metrics Shared Task. In *Proceedings of the Eighth Workshop on Statistical Machine Translation*, pages 45–51, Sofia, Bulgaria, August. Association for Computational Linguistics.

Gideon Maillette de Buy Wenniger and Khalil Sima'an. 2011. Hierarchical Translation Equivalence over Word Alignments. In *ILLC Prepublication Series, PP-2011-38*. University of Amsterdam.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of ACL'02*, pages 311–318, Philadelphia, PA, USA.

Miloš Stanojević and Khalil Sima'an. 2014. BEER: BEtter Evaluation as Ranking. In *Proceedings of the Ninth Workshop on Statistical Machine Translation*, pages 414–419, Baltimore, Maryland, USA, June. Association for Computational Linguistics.

David Talbot, Hideto Kazawa, Hiroshi Ichikawa, Jason Katz-Brown, Masakazu Seno, and Franz Och. 2011. A Lightweight Evaluation Framework for Machine Translation Reordering. In *Proceedings of the Sixth Workshop on Statistical Machine Translation*, pages 12–21, Edinburgh, Scotland, July. Association for Computational Linguistics.

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

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.

Hao Zhang, Daniel Gildea, and David Chiang. 2008. Extracting synchronous grammar rules from word-level alignments in linear time. In *Proceedings of the 22nd International Conference on Computational Linguistics-Volume 1*, pages 1081–1088. Association for Computational Linguistics.